TÉCNICO LISBOA

REPOX: Obtenção e agregação de dados para indexação em bibliotecas digitais em rede

Tiago João de Sousa Marques

Dissertação para obtenção do Grau de Mestre em

Engenharia Informática e de Computadores

Júri

Presidente:	Prof. Luís Eduardo Teixeira Rodrigues
Orientador:	Prof. José Luís Brinquete Borbinha
Co-orientador:	Prof. Luís Manuel Antunes Veiga
Vogal:	Prof. Pável Pereira Calado

Outubro 2012

Agradecimentos

Gostaria de agradecer aos meus pais e à minha companheira por todo o apoio que me deram durante esta fase tão decisiva na minha vida. Gostaria igualmente de agradecer ao Eng. Gilberto Pedrosa por toda a ajuda e todos esclarecimentos feitos sobre os assuntos relativos à framework do REPOX e às bibliotecas digitais. Finalmente gostaria de agradecer ao Eng. João Edmundo pela ajuda no desenvolvimento da interface gráfica da solução.

Abstract

With the advent of the Digital Libraries and Archives came the need to share with the community information that each of these entities has. This reality forced the creation of solutions that enable the dissemination of information present in each of these entities, so that it could be searched quickly and easily. However, due to the growth of new technologies, new goals were set by the entities listed above, passing by the possibility of researching information in the texts of the documents. Thus, this thesis will be focused on the demand for existing technologies and to obtain data synchronization and how these can be applied to solve facilitate the collection of new data content, for future indexation in search engine.

Keywords

Digital Libraries, metadata, REPOX, content harvest (fulltext).

Resumo

Com o aparecimento dos Arquivos e das Bibliotecas Digitais surgiu a necessidade de partilhar com toda a comunidade a informação que cada uma destas entidades possui. Esta realidade obrigou à criação de soluções que permitissem a divulgação da informação presente em cada uma destas entidades, de forma a esta puder ser pesquisada de forma simples e rápida. No entanto, devido ao crescimento das novas tecnologias, foram estabelecidos novos objectivos pelas entidades supracitadas, que passam pela possibilidade de pesquisa de informação presente nos textos dos documentos. Assim, esta tese vai-se focar na procura de tecnologias existentes para obtenção e sincronização de dados e de que forma estas podem ser aplicadas para resolver facilitar a recolha dos novos conteúdos de dados, para uma futura indexação num motor de pesquisa.

Palavras-chave

Bibliotecas digitais, metadados, REPOX, recolha de conteúdos (fulltext).

Índice

Índice de figuras

Índice de tabelas

Lista de Abreviaturas

Abreviação	Definição
EuDML	European Digital Mathematics Library
FTP	File Transfer Protocol
GWT	Google Web Toolkit
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
OAI-PMH	Open Archives Initiative Protocol for Metadata Harvesting
UI	User Interface
XHTML	Extensible HyperText Markup Language
XML	Extensible Markup Language
Xpath	XML Path Language

1. Introdução

1.1 Motivação

O aparecimento de bibliotecas e arquivos digitais, gerou um grande interesse na partilha da informação descritiva dos registos existentes, por partes das mesmas em projectos internacionais, como a Europeana[1], TEL[2] e EuDML[3]. Estas entidades têm como objectivo a recolha desta informação descritiva, de forma a facilitar o seu processo de pesquisa, criando as condições necessárias de acesso dos mesmos aos utilizadores, sem que estes necessitassem de analisar todas as fontes de informação separadamente. Estas iniciativas vieram desta forma colmatar uma lacuna correspondente à inexistência de um ponto centralizado de pesquisa, facilitando o acesso à informação por parte de investigadores, alunos ou de qualquer outro utilizador que assim o deseje.

Tendo em conta os cenários descritos, as organizações dispostas a partilhar os dados muitas vezes encontram dificuldades em fazê-lo, devido aos seus sistemas informáticos não suportarem de forma nativa o protocolo OAI-PMH[1], que é um requisito comum associado aos projectos internacionais supracitados. Esta tecnologia é suportada por várias soluções, tanto comerciais como open source, mas em muitos casos é difícil fazer os investimentos necessários para a compra de soluções pagas, ao passo que a utilização de software open source normalmente necessita de alguma adaptação do mesmo à realidade da entidade que o vai utilizar. Na maioria dos casos estas não possuem, nos seus quadros de trabalhadores, as capacidades técnicas para efectuar essas alterações o que poderá novamente implicar investimentos.

Assim, o OAI-PMH é um protocolo baseado na arquitectura Cliente-Servidor, usando XML sobre HTTP. Este assenta, como pretendido na conferência de Santa Fé[2], na distinção clara entre o que são **Data Providers** e **Service Providers**.
Data providers são as entidades que possuem informação (metadados) e estão dispostas a partilhá-las com os outros. Estas disponibilizam a informação sem qualquer tipo de custos, sendo que podem mesmo oferecer acesso a outros tipos de conteúdos, como textos ou imagens, mas que não tratados por este protocolo.

Service providers são as entidades que agregam a informação proveniente dos Data providers e a disponibilizam a toda a comunidade, através da introdução de serviços que permitem visualizar a informação a um nível mais alto (motores de busca ou browsers por exemplo).

Com o intuito de facilitar a partilha de dados entre as bibliotecas digitais e os projectos que promovem a agregação destes mesmos, foi criada uma plataforma open source de nome REPOX. Esta é uma framework que

[1] Europeana –The European Digital Library -http://dev.europeana.eu/
[2] TEL - The European Library- http://www.theeuropeanlibrary.org
[3] EuDML– European Digital Mathematics Library- http://www.eudml.eu/

visa simplificar os processos de partilha e recolha de dados a qualquer uma das entidades mencionadas anteriormente, pois possui uma instalação e configurações fáceis, diminuindo assim o esforço a níveis técnicos que possa ser requerido, aumentando em muito a simplicidade de iniciação na partilha e recolha dos dados.

Esta framework de gestão de metadados dispõe actualmente de tecnologias para:

- Aquisição e armazenamento de metadados provenientes de diferentes fontes, utilizando os seguintes protocolos HTTP, FTP, o OAI-PMH e o Z39-50;
- Transformação de metadados de acordo com especificações, regras e modelos de cada entidade;
- Apresentação e disponibilização de informação adquirida.

No entanto com o crescimento das capacidades e dos recursos informáticos, têm surgido novos cenários de partilha e recolha de metadados, nomeadamente a transferência não só dos metadados produzidos pelas bibliotecas e arquivos digitais, mas também os conteúdos referenciados por estes mesmos dados, como é o caso de imagens ou documentos dos mais variados tipos (ex: artigos científicos, jornais, revistas, etc), ainda vídeos ou áudio. Estes são novos desafios que abrem um novo conjunto de requisitos que irão impor suporte a novos processos de recolha de agregação de informação.

1.2 O Problema

Actualmente existem várias entidades dispostas a partilhar os seus recursos catalogados através de metadados, sendo exemplos disso o projecto Europena initiative[4], bibliotecas, museus e arquivos. Os objectivos principais que levam ao interesse na partilha de recursos passam pela preservação dos mesmos e criação de estruturas de pesquisa de dados.

Em termos da preservação dos dados é importante ter mais do que uma cópia dos dados, daí surge a necessidade de um **sistema de sincronização de dados** entre as entidades que irão fazer a sua manutenção e preservação das fontes de dados, de modo a tornar este processo mais simples e automatizado.

Relativamente à criação de estruturas de pesquisa de dados, estas têm por objectivo a utilização dos conteúdos de forma a torná-los pesquisáveis, sejam estes provenientes dos metadados, ou dos objectos referenciados pelos metadados (especialmente fulltext), que introduz desafios os quais se pretende estudar ao longo desta tese.

A escolha das recolhas ser feita particularmente sobre o fulltext, parte do facto de estes serem objectos digitais, constituídos essencialmente por texto, o que favorece a criação de estruturas de pesquisa de dados extremamente ricas, pois comparativamente à informação existente nos metadados a quantidade de informação disponível para enriquecer a pesquisa dos dados, partindo de um objecto de fulltext é massiva, criando assim óptimas condições para a eficácia e qualidade das pesquisas.

[4] Europeana - http://www.europeana.org/

1.3 Questões de Investigação

Com base na análise feita anteriormente ao problema é possível inferir as seguintes questões, que serão a base para esta tese.

1. Que técnicas são mais eficientes para a recolha e sincronização de objectos de conteúdos referenciados pelos metadados (especialmente fulltext), considerando tanto os cenários de uma primeira recolha como os de sincronização futura em caso de alterações?
2. Será possível optimizar o processo de recolha e sincronização de metadados, passando este a ser mais rápido e eficiente que o processo omissão com os servidores e clientes OAI-PMH actuais?

1.4 Objectivos

Os principais objectivos para esta dissertação são:

- Desenvolvimento de uma solução tecnológica que faça a recolha e o armazenamento de conteúdos referenciados pelos metadados no âmbito das bibliotecas digitais;
- Introdução de sistemas de sincronização de objectos referenciados pelos metadados e dos próprios metadados, de forma a reduzir o custo de obtenção e/ou actualização da informação relativa aos mesmos;
- Criação de um ambiente gráfico adequado, que permita o controlo do progresso das tarefas;
- Criação de uma arquitectura extensível, com capacidade de interligação com o projecto REPOX, por forma a flexibilizar a alargar o âmbito de soluções para recolha de dados que este já fornece.

Em suma, esta investigação procura desenvolver uma solução que adicione capacidade de recolha e sincronização de múltiplos tipos de informação em cenários de interoperabilidade entre sistemas e agregação de dados no âmbito das bibliotecas digitais.

1.5 Estrutura do documento

O resto do documento encontra-se organizado da seguinte forma: o Capítulo 2 apresenta algumas tecnologias relacionadas com o tema da tese, revendo o que existe no âmbito dos campos da replicação e sincronização de dados. O Capítulo 3 apresenta uma análise mais aprofundada do problema, delineando os limites das soluções actuais e os desafios que se pretende resolver. No Capítulo 4 é apresentada a Solução escolhida, onde é dada uma visão global da mesma, demonstrando a arquitectura e o funcionamento da tecnologia associada a recolha do fulltext. De seguida, no Capítulo 5, é apresentada a implementação da solução, sendo que no Capítulo 6 é feita uma avaliação da mesma através de utilização da mesma em cenários reais. Finalmente e para terminar a dissertação são apresentadas as conclusões e o trabalho futuro que possa existir no Capítulo 7.

2. Estado da Arte

Tendo em conta os objectivos definidos no âmbito das bibliotecas digitais e dos projectos de agregação de dados e as questões levantadas na secção anterior, é possível definir quais as soluções existentes que possam ajudar a solucionar as mesmas. Assim para as questões relativas à possibilidade de optimizar o processo de recolha e sincronização de metadados e para a questão sobre as técnicas de recolha e sincronização de objectos (fulltext) pretende-se apresentar algumas soluções focando os seguintes temas: Sincronização de Dados, Sincronização de Ficheiros e Data-sharing Middleware.

2.1 Sincronizadores de Dados

Nesta secção irão ser abordadas as tecnologias relativas à sincronização de dados e que possam relevantes no âmbito do REPOX.

2.1.1 SyncML

O SyncML[3][5] (Synchronization Markup Language) é um protocolo de sincronização de informação independente da plataforma ou dispositivo utilizado, definido pela Open Mobile Alliance[6]. O principal objectivo deste protocolo passa por ser a facilidade e eficiência da sincronização de dados, sendo principalmente utilizado para sincronizar email, calendários, contactos, etc. entre diversos tipos de dispositivos, móveis (PDA, computador portátil) ou fixos (PC ou servidor por exemplo). Esta iniciativa está associada a algumas das maiores empresas do ramo das telecomunicações e software como a Ericsson, Nokia, IBM, Motorola e a Symbian, assim como algumas empresas de comunicações wireless.
A arquitectura do SyncML baseia-se em dois objectivos :

- Sincronização de dados presentes em rede com os dados de qualquer dispositivo móvel;
- Sincronização de dados presentes em dispositivos móveis com dados existentes em rede.

Estes objectivos obrigaram a que este protocolo tivesse características muito bem definidas, sendo estas claramente as mais-valias presentes no SyncML:

- Funcionamento através de redes wireless e redes físicas;
- Suporte a múltiplos protocolos de transporte de dados, como o HTTP, WAP, SMTP, etc;
- Capacidade de transporte de dados independente do formato;

[5] The SyncML Initiative – http://xml.coverpages.org/syncML.html
Synchronization Markup Language – http://wiki.horde.org/SyncML/
http://www.openmobilealliance.org/tech/affiliates/syncml/syncmlindex.html
http://developers.sun.com/mobility/midp/articles/syncml/

[6] http://www.openmobilealliance.org/AboutOMA/Default.aspx

- Permissão de acesso de acesso aos dados por diferentes tipos de aplicações;
- Cuidado com as limitações dos dispositivos móveis, nomeadamente ao nível da quantidade de dados que estes podem albergar.

Em conclusão o SyncML é direccionado especialmente para os requisitos presentes no "mundo wireless", pois este minimiza o uso de largura de banda necessária para a transferência de dados e consegue lidar com desafios de sincronização associados a redes de baixa qualidade e de alta latência.

2.2 Sincronizadores de ficheiros

Nesta secção irão ser abordadas as tecnologias relativas à sincronização de ficheiros, que possam ser relevantes para o tópico em estudo ao longo desta tese.

2.2.1 Rsync

O Rsync[4] é um programa open source, desenhado para funcionar em qualquer plataforma, seja ela Windows, Linux ou Mac. Esta aplicação tem como principal funcionalidade a sincronização de pastas e de ficheiros, que estejam presentes em localizações diferentes. Esta tecnologia foi desenvolvida com o seu foco nas redes de baixa capacidade, onde a largura de banda é reduzida e existe uma elevada latência na transmissão dos ficheiros. Para isso utiliza formas para reduzir a quantidade de informação que é transmitida, cumprindo assim os requisitos a que se propõe.

O funcionamento deste sistema começa pela subdivisão de um ou mais ficheiros em pedaços que não se sobreponham de tamanho fixo (que podem ir de 500 a 1000 bytes), sendo que os últimos blocos poderão ter um tamanho inferior ao tamanho dos blocos utilizados. Estes blocos passam por uma fase onde são submetidos ao cálculo de checksums. De cada um destes blocos é obtido um checksum mais fraco de 32 bits, também conhecido por Rolling Checksum e um segundo, mais forte, de 128 bits o MD5 checksum. Em fases iniciais do projecto foi utilizado o MD4, tendo sido substituído posteriormente pelo MD5. Terminada a fase de obtenção dos valores do checksums de cada bloco, estes são enviados a outro computador que contem informação que necessita de ser actualizada, onde estes são comparados com os checksums fracos e fortes dos blocos dos ficheiros existentes nesta localização. Desta comparação será recolhida informação que irá servir para a fonte produzir uma solução (conjunto de instruções) que permita a actualização da informação no local onde se encontram as réplicas. As instruções podem ser constituídas ou por referências a blocos de informação ou por blocos de informação mesmo, sendo esta última apenas enviada quando é necessário inserir novos dados nos ficheiros.

É possível ainda através do Rsync fazer pequenas configurações no processo de sincronização de dados para que possam melhorar o seu funcionamento. Algumas destas passam pela possibilidade de comprimir e descomprimir blocos que irão ser enviados e ainda de cifrar a informação enviada com uso de protocolos como o SSH. É possível através das hipóteses de configuração do rsync de limitar a largura de banda utilizada pelo programa,

controlando assim todo o fluxo de dados e permitindo adaptar o funcionamento desta software as condições que as ligações entre os utilizadores possam apresentar.

Em conclusão pode ser verificado que este sistema de sincronização de informação apresenta grandes benefícios para os seus utilizadores pois, como já foi demonstrado anteriormente. Este apresenta uma grande flexibilidade e adaptação a vários meios e condições de partilha de informação, devido às características que o definem, sendo a principal a baixa quantidade de informação que este necessita de enviar aquando de uma actualização de dados. Devido a estas particularidades esta tecnologia é muito utilizada em vários contextos, tendo sido reaproveitada para vários outros projectos como poderá ser visto ao longo deste documento.

De referir ainda a existência de algumas variações da implementação do rsync, sendo feitas algumas alterações tendo em vista casos mais específicos de utilização do mesmo sistema, mas sempre com o objectivo de o tornar mais eficiente em todos os aspectos. Assim algumas destas são o In-Place Rsync [5] e Multiround Rsync [6].

2.2.2 Unison

Unison[7] é um sincronizador de ficheiros open source, que um pouco à semelhança do SyncML, tem como principais objectivos ser portável, estável e robusto e não menos importante, a sua utilização ser transversal em vários sistemas operativos e arquitecturas, como são casos o Unix e Windows. Este facto permite que seja possível que num servidor com Windows esteja sincronizado com um computador portátil com um sistema Unix. A sincronização é feita através do algoritmo de rsync, definido na secção anterior, sendo este usado para evitar desperdícios no envio de informação. Este apenas envia partes dos ficheiros que necessitam de ser actualizados e não todo o seu conteúdo, aumentando assim a velocidade de sincronização.

Esta ferramenta apresenta um conjunto de características muito interessantes que a tornam única:

- Sincronização de ficheiros entre diferentes plataformas. Problema com a utilização de nomes que em diferentes sistemas podem ser considerados ilegais;
- Capacidade de restauro de documentos por utilização de um sistema de cópias. O número de cópias é limitado para poupar espaço de armazenamento;
- Detecção de conflitos, em casos de alterações do mesmo ficheiro em fontes diferentes, sendo o utilizador informado do acontecimento;
- Possibilidade de resolução de conflitos através da chamada de aplicações externas;
- Sincronização entre duas ou mais máquinas feita por TCP/IP, o que permite a utilização de sockets ou o protocolo SSH, sendo o segundo uma fonte de comunicação mais segura e por a isso a aconselhada para o efeito;
- Robustez e resistência a falhas de comunicação ou crash de sistema.

2.2.3 SyncToy

O SyncToy[7] é uma ferramenta desenvolvida pela Microsoft, para os seus sistemas operativos Windows XP, Vista e 7, com o intuito de automatizar a sincronização de ficheiros e pastas num mesmo computador, num dispositivo externo (pen usb por exemplo) ou ainda numa outra máquina existente na mesma rede. Uma utilização típica desta aplicação passa pela partilha de ficheiros, como fotografias ou músicas e criação de cópias de segurança de dados presentes num computador.

O funcionamento deste sistema passa normalmente pela sincronização de duas pastas (pasta da esquerda e da direita) que como foi dito anteriormente podem estar alocadas em diferentes dispositivos e localizações de rede. Assim são disponibilizadas três hipóteses de métodos de sincronização:

- Synchronize: este método certifica-se que ambas as pastas têm os mesmos ficheiros, o que pode implicar cópias, remoções e alterações de nomes a ficheiros em ambas as pastas, garantindo que um estado de igualdade entre estas seja atingido;
- Echo: este método preocupa-se em procurar as diferenças existentes entre a pasta da direita e a da esquerda, alterando posteriormente a pasta do lado direito para garantir a sincronização das mesmas;
- Contribute: este processo é em tudo similar ao método anterior, sendo que se distinguem por este não propagar remoções de ficheiros para a pasta da direita, preocupando-se apenas com a adição de novos ficheiros e de alterações de nomes de ficheiros existentes.

Este sistema de sincronização permite ainda fazer uma previsão do estado final por aplicação da sincronização, sem realmente ter feito a sincronização, permitindo verificar as modificações propostas, dando a possibilidade de alterar as acções que estas iriam ser executadas previamente à execução das mesmas, diminuindo assim o risco de perda de informação. De referir ainda que aquando de uma sincronização com remoção de ficheiros, é possível configurar o SyncToy para que este envie os ficheiros removidos para a Reciclagem do sistema, aumentando assim a segurança no tratamento dos dados e sendo esta mais uma técnica importante para a prevenção de perdas de informação.

Em termos de resolução de conflitos, a forma como estes são resolvidos depende das acções definidas no momento da criação do par de pastas que é necessário sincronizar, sendo passíveis alguns cenários de resolução por estas acções a alteração do nome de um ficheiro em ambas as pastas, a remoção do ficheiro numa pasta e a alteração do seu nome noutra ou ainda a mudança de nome de um ficheiro numa pasta e a sua alteração noutra.

Para finalizar o processo de sincronização entre duas pastas este sistema promove ainda a captura de uma imagem final das pastas (snapshot), que irá conter informação relativa aos ficheiros que integram as pastas, os seus tamanhos, datas de alteração, etc, o que irá permitir uma maior facilidade na previsão por parte do SyncToy aquando da análise das diferenças e de como lidar com as sincronizações que são necessárias.

[7] http://www.microsoft.com/download/en/details.aspx?DisplayLang=en&id=8358
http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=15155

Em conclusão este sistema é muito direccionado para os requisitos ambicionados por utilizadores que trabalham com imagens, podendo também ser usado para a gestão de ficheiros em pastas. É um sistema com algumas limitações no que respeita à resolução de conflitos e não permite a passagem de ficheiros entre partilhas o que o torna de alguma forma limitado quando comparado com outros softwares concorrentes.

2.2.4 Sync Center

O Sync Center[8] é um sistema de sincronização de ficheiros presente no sistema operativo desenvolvido pela Microsoft, estando integrado nos dois mais recentes projectos desta empresa o Windows Vista e o Windows 7.

Este sistema tem como principal objectivo a sincronização de ficheiros e pastas com dispositivos móveis, pastas presentes numa rede ou ainda com outros programas, sendo este um agregador de informação que gere todos dispositivos que se sincronizam com o dispositivo que albergue este software.

O funcionamento do Sync Center tenta ser o mais user-friendly possível, visto este estar incorporado num sistema operativo utilizado pelas massas, apresentando apenas ao utilizador a escolha da localização ou dispositivo onde se encontram os dados, quais os dados que se deseja sincronizar e os períodos em que se deve efectuar essa mesma sincronização. Este sistema oferece assim uma plataforma onde os utilizadores podem aceder a todos os dispositivos e aos ficheiros presentes nos mesmos, controlar a sincronização dos mesmos e resolver conflitos que possam advir do processo anterior.

Uma das principais inovações presentes neste sistema são os "Offline Files". Estes permitem que sejam designadas pastas e ficheiros as quais vão estar disponíveis para ser acedidas e alteradas mesmo quando não existe uma ligação entre os dispositivos que as mantêm. Isto permite a alteração dos dados em qualquer altura, sendo que quando existe uma ligação com a versão presente online esta será sincronizada e actualizando ambas as versões para o mesmo estado. Este processo pode levar a existência de conflitos que normalmente são resolvidos com a escolha da versão mais recente dos dados gerados. Caso os dados tenham sido alterados em ambas as localizações é dada ao utilizador a responsabilidade da escolha da versão que se deseja manter, sendo possível mesmo guardar as duas versões.

Para concluir o Sync Center é uma boa plataforma para unificar todos os processos de sincronização que possam advir dos mais variados dispositivos, indo desde uma simples pen usb ou um leitor de música portátil a uma rede onde se encontram os ficheiros partilhados. Este sistema apresenta algumas lacunas nomeadamente ao nível da informação que transmite ao utilizador sobre as actualizações que são feitas sobre os dados, sendo que guarda apenas uma versão de cada documento, o que conduzir a problemas de perda de informação proveniente de uma

[8] http://windows.microsoft.com/en-US/windows-vista/How-to-keep-your-information-in-sync
http://msdn.microsoft.com/en-us/library/aa369140%28v=vs.85%29.aspx

resolução de conflitos que possa não ter sido tão satisfatória.

2.2.5 DropBox

Dropbox[9] é um sistema de armazenamento online de ficheiros, acessível a partir de qualquer computador, independentemente do sistema operativo que possua. Actualmente este é suportado em sistemas como Windows, Linux, Max OS X, e ainda em dispositivos móveis que possuam Android, Windows Phone e ainda em Iphones, Ipad ou BlackBerrys. Este sistema inicialmente fornece aos utilizadores uma capacidade de 2GB de capacidade de armazenamento que pode chegar aos 8 GB. Para utilizadores que escolham pagar para tirar um maior proveito deste serviço este pode chegar até aos 100GB.

O funcionamento deste sistema passa por ser bastante simples e fácil de perceber por parte do utilizador. Existem duas formas de funcionar com esta tecnologia, sendo que a mais recorrente passa por instalar num computador ou dispositivo móvel a aplicação da dropbox. Esta cria uma pasta no sistema onde está instalada e os utilizadores apenas têm de ir colocando nessa pasta os ficheiros que desejem fazer uma cópia dos mesmos. Estes serão automaticamente guardados nos servidores deste serviço e replicados em todos os dispositivos que tenham a mesma conta associada, sendo possível ter múltiplos sincronizados com a mesma dropbox. No caso de ficheiros serem adicionados, removidos ou alterados nesta pasta, todas estas alterações são propagadas para o Servidores deste serviço e o utilizador é avisado das alterações que estão a proceder, sendo todos os dispositivos automaticamente ligados pela mesma conta actualizados. A outra forma é através de um qualquer browser, onde a partir de qualquer computador com acesso à Internet é possível fazer todas as acções supracitadas, pois basta aceder à pagina da Dropbox fazer o login e todos os dados estão disponíveis ao utilizador, podendo este proceder as alterações que desejar, que a semelhança do que foi dito anteriormente todas serão propagadas pelos seus dispositivos, caso estes estejam conectados à Internet.

Outra possibilidade interessante fornecida por este serviço é a capacidade de partilhar pastas com outros utilizadores, permitindo assim trabalho colaborativo entre um ou mais utilizadores, sem requerer a presença física dos mesmos. Este sistema apresenta ainda um sistema de versões associado a todos os ficheiros presentes na Dropbox. Este sistema permite o retrocesso a versões antigas de ficheiros em caso de necessidade, sendo que o histórico de cada ficheiro apenas tem uma duração de 30 dias para as versões livres, enquanto nas contas com versões pagas a duração pode ser ilimitada. Outra limitação das contas livres encontra-se no tamanho máximo que cada ficheiro pode ter, sendo que cada um não pode ultrapassar os 300 MB, o que novamente não acontece nas versões pagas do mesmo serviço.

As maiores desvantagens que podem ser apontadas a este serviço passam por ser:

- Em caso de alteração simultânea do mesmo ficheiro a Dropbox não possui a capacidade para resolução de conflitos, dando apenas um aviso ao utilizador da presença do mesmo e em que ficheiro, sendo

[9] https://www.dropbox.com/

criadas duas versões do mesmo ficheiro com as respectivas diferenças e deixando à responsabilidade do utilizador a resolução do problema;

- Existência de algumas questões relativas à privacidade dos dados que os utilizadores guardam nos servidores da Dropbox, pois as técnicas de conservação e compressão dos dados utilizadas por esta empresa têm sido postas em causa, por poderem descodificar a informação que os utilizadores guardam invadindo assim a sua privacidade.

2.2.6 Live Mesh

Live Mesh[10] é um sistema desenvolvido pela Microsoft, sendo em múltiplos aspectos similar à DropBox, sendo que o seu objectivo principal acaba por ser guardar ficheiros online e garantir a sua sincronização. À semelhança da DropBox também é disponibilizado espaço para livre utilização sendo o valor inicial deste de 5GB.

Actualmente as semelhanças entre este serviço são bastantes, sendo que algumas das diferenças se baseiam no facto de este serviço apesar ser utilizável em várias plataformas como o Windows ou o Mac OS X, deixa de fora o mundo open source sendo que os utilizadores de Linux ou sistemas como o Android não poderam beneficiar da utilização de um sistema desta índole, o que pode estar implícito em algum desconhecimento da existência desta tecnologia. Nestes casos é oferecida apenas a possibilidade de acesso aos ficheiros via web.

Esta tecnologia acaba por ser base de algumas das funcionalidades apresentadas nos dias de hoje na Dropbox, especialmente a utilização do browser como ferramenta de aquisição, alteração ou remoção de ficheiros.

Em suma esta tecnologia é em múltiplos aspectos similar à Dropox, acabando por se diferenciar na capacidade de armazenamento disponibilizada no serviço livre de pagamentos e na inexistência do seu suporte em tecnologias open source.

2.2.7 Syncany

Syncany[11] é um novo software de sincronização de ficheiros. Este em comparação com o Live Mesh da Microsoft ou a Dropbox apresenta algumas vantagens relativamente aos seus concorrentes, nomeadamente o facto de ser um produto open source, logo possibilitando a facilidade de obtenção do código do mesmo e a sua reutilização. Outra vantagem passa pela capacidade de encriptar os ficheiros presentes, na máquina em que estão a ser partilhados, garantindo desta forma a segurança dos mesmos e a privacidade que normalmente os utilizadores desejam num serviço desta natureza. Este serviço tem o objectivo de ser extensível através de introdução de plugins, para que possam ser introduzidos novos protocolos para sincronização de dados. Actualmente os protocolos suportados são FTP, Box.net, Amazon S3, Google Storage, Imap, etc sendo objectivo dos criadores desta tecnologia introduzir no futuro suporte a outros protocolos como o Windows Share.

[10] http://explore.live.com/windows-live-mesh-devices-sync-upgrade-ui
http://en.wikipedia.org/wiki/Windows_Live_Mesh

[11] http://www.webupd8.org/2011/05/syncany-great-dropbox-alternative-which.html

Apesar de este sistema apresentar algumas ideias revolucionárias, acaba por ser muito imberbe, especialmente por apenas funcionar em sistemas Linux e se encontrar em pleno estado de desenvolvimento. Infelizmente um dos objectivos presentes passa por nunca ser suportado por plataformas como o Windows o Mac, o que lhe pode vir a retirar alguma visibilidade e reconhecimento no mundo da informática.

2.3 Software de controlo de versões

Este tipo de software não se enquadra no tipo de utilizadores que normalmente utilizam sincronizadores de ficheiros, visto que um software de controlo de versões é utilizado com mais regularidade para o desenvolvimento de projectos onde existe a partilha de ficheiros entre vários responsáveis pela evolução do mesmo. Assim sendo, o objectivo do estudo deste tópico passa pela análise de uma tecnologia chamada Git, que tem como principais características velocidade, eficiência e escalabilidade. Estas são alguns dos objectivos que se pretendem melhorar no REPOX com a execução desta tese.

2.3.1 GIT

Git[12] é um sistema open source de controlo de versões distribuído. Este, ao contrário do que acontece noutros tipos de sistemas de controlo de versões, não utiliza um repositório central como acontece por exemplo no CVS[13], onde são guardados todos os ficheiros e a sua evolução ao longo do desenvolvimento do projecto e os utilizadores acedem para efectuarem operações sobre o mesmo. Com o sistema Git cada projecto é um repositório com todas as capacidades de controlo de versões e de históricos assegurados. Este sistema permite ainda uma fácil ramificação dos projectos, podendo estes ser locais ou remotos podendo ser mesmo inseridos noutros projectos, sendo cada ramo uma cópia exacta do repositório de onde deriva. Este processo permite que não haja perdas de informação pois deixa de existir um repositório para passar a haver N.

Uma grande vantagem inerente ao facto de cada utilizador ter um repositório próprio onde efectua o desenvolvimento é a inexistência de necessidade de conexão à Internet para que este registe alterações no projecto ou permita a procura de alterações antigas ou ainda permita a fusão de código proveniente de ramos locais ao sistema. A conexão ao exterior é apenas necessária para partilhar ou obter dados de ramos remotos de um projecto.

O sistema Git é transversal aos protocolos de Internet principais, sendo possível que os repositórios sejam partilhados tanto por HTTP, FTP, rsync ou ainda por ssh, o que facilita a sua utilização e aplicação nos mais diversos contextos. Visto a sua implementação ser maioritariamente em C permitiu que esta plataforma seja também transversal ao sistema operativo em que se trabalhe, sendo mesmo utilizado como ferramenta complementar a sistemas de IDE como o Eclipse, IntelliJ ou NetBeans.

[12] http://git-scm.com/

[13] http://cvs.nongnu.org/

Como sistema de armazenamento, o Git usa um sistema de snapshots do estado de toda a árvore de ficheiros. Isto passo é efectuado a cada nova alteração efectuada no projecto. Numa fase inicial do projecto, este utilizava um sistema de deltas que consistia em guardar as diferenças existentes entre os ficheiros existentes aquando da submissão de alterações. Este processo revelou-se muito dispendioso ao nível do espaço necessário para o armazenamento dos mesmos, sendo necessário passar para um sistema de snapshots do estado de todos os ficheiros, obtendo-se assim melhorias ao nível tanto do armazenamento como da eficiência da obtenção das diferenças submetidas e ainda na pesquisa e partilha das mesmas com outros ramos do projecto.

Tendo já mencionado como característica deste sistema a facilidade da partilha de dados entre os diversos ramos do projecto, este processo torna-se mais cómodo devido ao Git possuir um sistema de detecção de conflitos, providenciando ferramentas que permitem aos utilizadores a visualização das diferenças e ajuda para a sua resolução. É ainda possível alterar as ferramentas usadas para este efeito, visto este sistema permitir a utilização de outras que sejam mais familiares aos utilizadores.

Assim e para concluir, podemos verificar que este sistema apresenta características que permitem facilitar principalmente equipas de desenvolvimento de projectos, pois reduz a complexidade na sincronização da informação gerada por diferentes intervenientes.

2.4 Data-sharing middleware

Como o próprio termo indica, middleware representa uma camada de abstracção entre dois sistemas utilizados. Neste caso representam sistemas que retiram a responsabilidade da tecnologia de obtenção dados a forma como estes são replicados para o sistema. A finalidade principal deste tipo de software é a de serem facilitadores de desenvolvimento de aplicações pois permitem a sua utilização sem grandes mudanças ao código das mesmas.

2.4.1 IceCube

O objectivo do projecto IceCube [8] é o fornecimento de uma plataforma que funcione como um conciliador de cópias do mesmo trabalho que sofreram alterações, para que a sua integração seja o mais pacífica possível. Este sistema está parametrizado para ter em atenção a semântica do tipo de dados que se está a trabalhar a aplicação dos mesmos ou do utilizador que gerou as alterações.

Este sistema tem como unidade base para o seu funcionamento o estado inicial de um ficheiro que se está a analisar e os registos (logs) gerados pelas acções aplicadas em cada réplica deste. Os registos (logs) fornecem ao sistema o histórico das acções tomadas por cada utilizador, permitindo assim aumentar as capacidades do programa, pois este fica a perceber as intenções de cada utilizador aquando da aplicação das alterações, facilitando a sua integração com as alterações de outros utilizadores. Ao contrário de outros sistemas o IceCube pretende reordenar as operações latentes nos registos (logs), para além da simples ordenação temporal, para assim procurar formas de minimizar os conflitos. Esta solução levanta uma questão que se prende com uma

possível explosão das combinações possíveis entre os registos (logs), tendo sido para isso aplicadas ao sistema sistemas de restrições (estáticas e dinâmicas) para controlar esta lacuna de desenho do sistema.

Restrições estáticas estão directamente relacionadas com a ordem de como são aplicadas as operações para chegar a um estado final. Apenas têm em atenção se estas são aplicadas de forma segura não se preocupando com estado actual dos objectos que estão a ser tratados.
Restrições dinâmicas pode ser ou operações ou pré-condições. Uma operação é um método que pode ou não modificar os objectos que estão a ser partilhados, indicando o sucesso ou insucesso da mesma. Uma pré-condição apenas verifica se estado actual de um objecto é válido.

O processo de funcionamento do IceCube passa por duas fases. Estas são:

- Execução isolada: nesta fase são aplicados os objectos partilhados um conjunto de actualizações produzidas pelo utilizador, sendo gerado um registo deste procedimento;
- Fase de reconciliação: nesta fase é feita a tentativa de sincronização de duas ou mais réplicas do mesmo objecto partilhado sendo subdividida em três novas fases;
    - Fase do escalonamento: nesta fase são criados escalonamentos a partir das várias combinações possíveis das actualizações. Estes escalonamentos são conjuntos de acções que irão ser aplicadas nos objectos partilhados respeitando as restrições estáticas para gerar um estado considerado correcto. Se o estado não for considerado correcto esse escalonamento é descartado, controlando assim a explosão que possa haver de combinações;
    - Fase da simulação: nesta fase são aplicados os escalonamentos de actualizações considerados válidos na fase anterior, verificando se as restrições dinâmicas são respeitadas, descartando novamente os conjuntos de actualizações que não respeitem as considerações definidas;
    - Fase da selecção: nesta última fase todas os escalonamentos que foram considerados válidos sendo estes comparados e classificados, sendo escolhido a solução que apresente o melhor resultado final. O resultado originário é posteriormente partilhado entre todas as réplicas do objecto para aplicação da solução obtida.

Em conclusão o IceCube é sistema de reconciliação de objectos partilhados, baseado nos registos gerados das actualizações produzidas pelos utilizadores. Esta solução tem como principal objectivo a diminuição de conflitos, sendo totalmente orientada para soluções de trabalho partilhado.

2.4.2 Semantic Chunks

Semantic chunks [9] são um conceito desenvolvido com o intuito de tentar resolver alguns problemas apresentados pelas técnicas (update-based e operational-based) utilizadas na base do trabalho cooperativo, nomeadamente na ajuda ao nível de fornecer garantias de consistência e de diminuição de conflitos aquando da sincronização de ficheiros.

A ideia por de trás deste conceito nasce de tentar adoptar as vantagens de cada uma das técnicas anteriormente mencionas e da utilização de chunks em LBFS e Haddock-FS, tendo dado origem a poupanças tanto de armazenamento de dados como de largura de banda utilizada na propagação das actualizações efectuadas nos projectos. O funcionamento deste sistema baseia-se na divisão de documentos em regiões semanticamente relevantes, que podem ser diferentes consoante do tipo ficheiro e a sua aplicação semântica. Estes pedaços de informação extraídos podem ir de um simples parágrafo num texto, a uma célula de uma folha da cálculo ou até mesmo a uma página de uma apresentação. Este facto promove a consistência de ficheiros num sistema, visto que as divisões dos mesmos ser feita semanticamente, tendo em conta o tipo de ficheiro, e não por blocos de tamanho constante como em outras soluções, reduzindo assim o número de conflitos provenientes da sincronização de ficheiros e aumentando a concorrência e a frequência de actualizações provenientes de múltiplas fontes.

Visto este ser um sistema direccionado para o trabalho cooperativo e apesar da redução de conflitos conseguida, estes continuam a existir e assim são utilizados vários esquemas para que os utilizadores os consigam resolver. Estes são:

- Votações para saber quais as actualizações que devem utilizadas;
- Actualizações adoptadas por decisão de utilizadores com maiores privilégios no desenvolvimento do projecto;
- Definição de períodos para a introdução de actualizações por parte de algum utilizador (lease);
- Partilha de informação com outros utilizadores sobre a alguma actualização que se pretende inserir no projecto.

Como conclusão pode-se observar que este sistema consegue obter as maiores virtudes das técnicas update-based e operational-based, reduzindo o número de conflitos produzidos por actualizações da informação e aumento da concorrência e frequência das actualizações. Existe ainda uma redução de largura de banda e da capacidade necessária para armazenamento devido a se lidar com fragmentos de informação facilitando o controlo das actualizações de ficheiros.

2.4.3 Xmiddle

O Xmiddle [10] é um sistema que tem como objectivo principal a partilha de informação entre dispositivos de computação móveis, como é o caso telemóveis, PDAs ou computadores portáteis, sem a existência de qualquer tipo de rede de comunicações fixa. Este sistema aborda principalmente cenários em que são utilizadas redes ad-hoc, mais especificamente comunicação entre apenas dois intervenientes.

Em termos de estruturação dos dados, este sistema guarda os mesmos em árvores de estruturas organizadas hierarquicamente, facilitando desta forma o acesso e manipulação dos mesmos. Toda a informação é representada internamente em XML para uma maior flexibilidade e facilidade de associação com o sistema supracitado.

Quanto ao funcionamento do Xmiddle, este pretende a sincronização de ramos de árvores que sejam comuns a dois dispositivos, criando assim duas novas árvores semelhantes e com a mesma versão nos dois dispositivos. Este começa por, quando há ligação entre dois dispositivos, verificar a existência de ramos partilhados entre estes. Confirmado este requisito o dispositivo que promove a conexão, que iremos chamar de D2, envia ao outro (D1) o histórico das alterações promovidas no ramo. Este verifica as diferenças entre os dois históricos e envia-as para D2 onde este irá gerar uma nova árvore baseada nas diferenças, promovendo assim uma fusão das duas árvores. Tendo terminado este processo D2 envia para D1o conjunto de modificações efectuadas para que o segundo possa gerar um nova árvore semelhante à de D2 e assim terminar o processo de sincronização entre ambos. Caso este tenha terminado com sucesso ambas as árvores irão apresentar a mesma versão, caso tenham havido conflitos aquando da fusão dos ramos o Xmiddle permite a definição de políticas de reconciliação de dados através de esquemas XML, que resolvam conflitos automaticamente quando estes existirem.

No caso de haver algum tipo de quebra da ligação, que impeça a continuação da comunicação das mudanças aplicadas nos dados, os dispositivos retêm as réplicas da última árvore estável que estava a ser partilhada, permitindo assim a evolução do desenvolvimento do trabalho apesar do acontecimento, sendo o processo de sincronização reinicializado aquando da existência de uma nova ligação entre os dispositivos em questão.

Para concluir pode-se afirmar que o desenvolvimento desta tecnologia tem como objectivo promover a descoberta de estratégias que lidem com os problemas associados à computação móvel como, perdas de conexão, baixa capacidade de largura de banda ou problemas energéticos.

2.4.4 Microsoft Sync Framework

O Microsoft Sync Framework[14] é uma plataforma desenvolvida pela Microsoft com o intuito de permitir uma fácil sincronização de qualquer tipo de documentos independentemente da aplicação, do tipo de dados ou de qualquer protocolo utilizado no desenvolvimento do projecto.

Para que seja possível a partilha de qualquer tipo de informação esta tem de estar armazenada em algum lugar. Nesta óptica a Microsoft decidiu definir actores a quem deu o nome de Participantes. Estes são os locais de onde se consegue obter a informação proveniente das fontes de dados, podendo estes serem um computador, uma pen drive, um PDA ou até mesmo um web service. Estes participantes podem ser de vários tipos, que variam conforme as suas capacidades de armazenamento e manipulação da informação tanto de forma local como remota e ainda da sua capacidade de executar aplicações para sincronização de informação directamente no dispositivo. Assim os participantes existentes são:

- Participantes Totais: são definidos por permitirem a criação de aplicações directamente nestes dispositivos bem como a definição da localização dos dados a serem guardados;

[14] http://msdn.microsoft.com/pt-pt/sync/default(en-us).aspx

- Participantes Parciais: são definidos pela sua capacidade de armazenamento de dados como exemplo disso pen drives ou SD cards;
- Participantes Simples: são definidos por apenas partilharem informação quando esta lhe é pedida, sendo exemplos disto os RSS feeds.

Para que os participantes possam partilhar dados entre si é necessária a existência de outra entidade. Esta é a chave de todo os sistema utilizado pelo Microsoft Sync Framework e dá pelo nome de Provider (fornecedor). Estes podem ser definidos pelos utilizadores, visto que os dados que serão partilhados podem ser de algum tipo não suportado por esta estrutura. No entanto são disponibilizados vários providers, sendo exemplo de alguns deles, providers para sincronização de bases de dados, de ficheiros e pastas, etc. Para cada fornecedor é especificado o tipo de dados que este irá sincronizar, sendo este responsável pela sua manutenção e garantia de consistência dos mesmos. O funcionamento destes vai para além das funções supracitadas, sendo que estes guardam informação relativa as alterações aplicadas aos dados, nomeadamente as mudanças ocorridas e o estado em que os dados se encontram. Esta informação é guardada num repositório de metadados que pode ser definido ou pelo criador do provider ou então utilizar o fornecido pelo próprio Microsoft Sync Framework.

O processo de sincronização deste sistema está directamente ligado a três módulos muito específicos. O Sync Provider, que fornece toda a comunicação entre todas as réplicas do projecto e ainda outros providers com se deseje comunicar com este e ainda o Data Source e o Metadata Store que armazenam respectivamente os dados e os metadados relativos aos dados.
Finalmente este sistema apresenta ainda um sistema de detecção e resolução de conflitos com mecanismos pré-definidos para a automática resolução dos mesmos. À semelhança do que acontecia com os providers, também é possível criar regras para resolução deste tipo de conflitos.
Em conclusão o Microsoft Sync Framework apresenta uma grande flexibilidade para a sua utilização, fornecendo assim uma plataforma avançada de utilização, sendo esta direccionada para utilizadores com conhecimentos mais avançados conseguindo assim partilhar e gerir a evolução dos seus projectos.

2.5 Análise

Tendo em conta as soluções que foram apresentadas nesta secção, é possível verificar que qualquer uma delas faz a replicação de ficheiros e pastas entre diferentes dispositivos garantindo a sua consistência. Assim neste subcapítulo iremos avaliar com base na tabela 1, como estas soluções podem ser utilizadas para resolver o problema enunciado no capítulo 1.

Analisando a tabela supracitada, podemos verificar que muitas das soluções apenas funcionam em determinados sistemas operativos, sejam eles proprietários ou livres, o que pode vir a ser um problema pois, é impossível, prever que sistemas operativos tanto os Data Providers como os Service Provider, o que limita automaticamente as escolhas para possíveis soluções dos problemas estudados. Outro factor de exclusão para o tipo de tecnologias que podem ser utilizadas como solução para os problemas desta tese é o tipo de licença que existe para

reutilização das tecnologias. Visto o REPOX não utilizar qualquer tipo de software proprietário esta terá de ser uma máxima que terá de ser mantida.

Passando para os critérios mais técnicos é necessário avaliar se as soluções estudadas têm a capacidade de lidar com algumas das especificidades dos cenários que são propostos pelos bibliotecas e arquivos digitais. Estes critérios são:

- Recolha de conteúdos partindo referencias fornecidas por metadados;
- Sincronização de actualizações de conteúdos recolhidos através objectos referenciados.

Tendo em conta estes dois critérios e considerando o estudo feito ao longo deste capítulo, que nenhuma das tecnologias tem a capacidade de efectuar recolhas de objectos referenciados, tanto em XML como noutros formatos, pois estas tecnologias são principalmente pensadas para garantir a actualização e a consistência dos dados que acolhem, onde estas têm acesso à localização original dos dados. Como é passível de observação esta é uma condicionante que exclui as tecnologias estudadas de serem uma solução viável para a resolução do problema em questão. Isso é igualmente visível para o caso da sincronização, pois mesmo que a recolha dos objectos fosse efectuada de forma separada, para permitir que as tecnologias fossem utilizadas apenas no cenário da sincronização, estas não possuem capacidade para enfrentar o desafio devido ao objectivo inicial a que estas se propõem, que exige que haja sempre de alguma forma uma ligação ao objecto original, não contemplando o caso de este estar apenas referenciado.

Assim é possível concluir que, com base nas tecnologias estudas e na especificidade dos cenários que são propostos a resolução, que a melhor solução passa pelo desenho e implementação de uma tecnologia que seja de alguma forma agregada à framework REPOX, que possua a capacidade de recolher a manter actualizados os objectos referenciados pelos metadados gerados pelas bibliotecas e arquivos digitais.

Tabela 1. Comparação de tecnologias de recolha e sincronização

	Plataforma	**Tipo de Sincronização**	**Serviço de sincronização**	**Agendamento**
SyncML	Independente	Online	Dados entre diversos dispositivos	Sim
Rsync	Windows, Mac OS X, Linux	Online	Ficheiros em redes de baixa capacidade	Usa OS
Unison	Windows, Mac OS X, Linux	Online	Algoritmo de Rsync	Usa OS
Synctoy	Windows	Offline	Pastas locais	Usa OS
Sync Center	Windows	Offline	Pastas locais	Não
DropBox	Windows, Mac OS X, Linux. Android, Windows Mobile, Iphone	Online	Pastas e ficheiros em múltiplos dispositivos	Não
Live Mesh	Windows	Online	Pastas e ficheiros em múltiplos dispositivos	Não
Syncany	Linux	Online	Pastas e ficheiros em múltiplos dispositivos	Não
Semantic -Chunks	Independente	Online	Ficheiros e redução de conflitos em redes de baixa capacidade	Sim
IceCube	Independente	Online	Reconciliador de actualizações de objectos partilhados	Sim
Xmiddle	Independente	Offline	Informação entre dispositivos móveis	Sim
Microsoft Sync Center	Microsoft	Online	Qualquer tipo de documentos independente da sua génese	Sim

3. Análise do problema

Neste capítulo vai ser analisada toda a problemática da tese, começando por analisar o que são bibliotecas digitais, seguindo-se uma demonstração dos cenários previstos para a implementação do novo paradigma de recolha de dados. Logo após será feita uma análise mais profunda ao OAI-PMH, para assim se perceber a tecnologia de transferência de dados mais utilizada no âmbito da disponibilização de dados pelas bibliotecas digitais. Para terminar irão ser descritas quais as principais questões resultantes da análise do problema e os desafios que se pretende resolver.

3.1 Bibliotecas Digitais

As bibliotecas digitais podem ser definidas como "uma colecção de objectos digitais, que inclui texto, vídeo e áudio, bem como métodos de acesso e obtenção de dados, e ainda para organização e manutenção de colecções de dados."[13] Esta definição permite perceber que as bibliotecas são entidades, que passaram de meros aglomeradores e organizadores de informação a "criadores" de objectos digitais, partindo dos registos que estas possuam no seu espólio, de forma a possibilitar não só uma nova forma de preservação dos mesmos, mas ao mesmo tempo uma nova forma de partilha destes conteúdos.

3.2 Metadados e objectos de digitais

A explosão do World-Wide Web possibilitou a múltiplos agentes, a disponibilização dos seus dados na Internet, de forma a estes serem pesquisáveis por todos os tipos de utilizadores. No entanto a tarefa de pesquisa de recursos que possuam relevância para a resolução de determinado problema pode ser morosa e muitas vezes pode implicar interacções entre várias fontes de informação, de forma a responder ao problema proposto. Para facilitar as tarefas supracitadas foi criado o conceito de metadados, em que os primeiros servem para caracterizar de forma explícita os últimos. No âmbito das bibliotecas e arquivos digitais este conceito é utilizado como descritor de informação dos recursos disponíveis, sendo que normalmente descreve o título, a data de criação, o autor, etc. do documento preservado por estas instâncias[15].

3.3 O Problema

Tendo em conta os objectivos definidos no Capítulo 1, é possível prever alguns cenários de interoperabilidade por parte das entidades que partilham a informação e as que fazem a recolha e agregação da mesma. É nesta aplicação que se conseguem vislumbrar os problemas que são a base para esta tese.

[15] http://dublincore.org/documents/2001/04/12/usageguide/generic.shtml

Assim os principais caso de uso que foram antecipados são a interoperabilidade entre Data Provider e Service Provider e a interoperabilidade entre Service Providers.

3.3.1 Interoperabilidade entre Data Providers e Service Providers

O normal funcionamento do sistema de agregação de dados, passa por um Data Provider publicar metadados para que o Service Provider equipado com a tecnologia REPOX possa fazer a recolha dos mesmos, e tendo em conta o novo paradigma previsto de recolha de conteúdos. Existem três cenários a ter em conta para esta secção. São eles:

- Interoperabilidade entre um Service Provider e um Data Provider que publica os metadados por OAI-PMH normal e os conteúdos por HTTP ou FTP;
- Interoperabilidade entre um Service Provider e um Data Provider que publica os metadados por OAI-PMH normal e os conteúdos por tecnologia recomendada pelo projecto REPOX;
- Interoperabilidade entre um Service Provider e um Data Provider que publica os metadados e os conteúdos por tecnologia recomendada pelo projecto REPOX.

3.3.2 Interoperabilidade entre Service Providers

Considerando desta vez cenários de interoperabilidade entre dois Service Providers em que um deles utiliza a tecnologia REPOX, passando a funcionar como agregador de dados, ou seja um Data Provider, existem novamente três cenários a ter em conta. São eles:

- Interoperabilidade entre um Service Provider e outro Service Provider que recolhe os metadados por OAI-PMH normal e os conteúdos por HTTP ou FTP;
- Interoperabilidade entre um Service Provider e outro Service Provider que recolhe os metadados por OAI-PMH normal e os conteúdos por tecnologia recomendada pelo projecto REPOX;
- Interoperabilidade entre um Service Provider e outro Service Provider que recolhe os metadados e os conteúdos por tecnologia recomendada pelo projecto REPOX.

3.3.3 Análise do problema

Analisando os cenários anteriores, é possível verificar que só em alguns deles é possível controlar a forma como como os dados são publicados e recolhidos, o que restringe de alguma forma o âmbito do problema, pois nesses casos é apenas possível fazer optimizações nas entidades que utilizem tecnologias recomendadas pelo projecto. Ainda assim é possível retirar alguns desafios interessantes, nomeadamente de como efectuar o processo de recolha de conteúdos de forma a evitar o download de colecções inteiras de ficheiros, quando existirem apenas pequenas actualizações.

Outra questão de valor pode passar pela optimização do processo de recolha dos metadados, mas para isso vai ser analisado um pouco mais a fundo o protocolo OAI-PMH.

3.4 OAI-PMH

O protocolo OAI-PMH é o resultado de um projecto desenvolvido pelo Open Archives Initiative (OAI), cuja sua principal actividade é centrada na resolução de problemas de interoperabilidade entre entidades, nomeadamente bibliotecas digitais e arquivos, pela criação e desenvolvimento de protocolos e standards para a disseminação de conteúdos[1].

A criação do OAI teve início em Outubro de 1999 na Convenção de Santa Fé[2] e as motivações existentes para esta foram :

- O rápido crescimento da Internet e a adopção das entidades escolares como meio de partilha de resultados;
- A morosidade tradicional da publicação do avanços feitos nos vários âmbitos escolares;
- A problemática da transferência de direitos dos autores para as editoras, introduzindo problemas de divulgação de resultados;
- Os atrasos ou a supressão de novas ideias provocados pelas revisões de pares, favorecendo em muitos casos publicações provenientes de instituições de maior prestígio, em detrimento de outras;
- Os preços das assinaturas por parte das bibliotecas tornaram-se demasiado dispendiosos.

Assim o objectivo inicial da OAI passava por alcançar repositórios de arquivos digitais de e-print que contivessem trabalhos de pesquisa guardados, tendo definido como propósito da convenção as seguintes predisposições:

- Criação de uma framework que permitisse a descoberta e partilha rápida dos conteúdos supracitados;
- Fornecimento de recomendações técnicas para a criação de arquivos que correspondessem as características do ponto anterior;
- Distinção clara entre o que são Data Providers e Service Providers.

Tendo observado a complexidade da problemática em análise, foi decidido que da convenção apenas iriam ser apresentadas soluções para a recolha de metadados, sendo estas as seguintes:

- Definição de um conjunto de elementos de metadados – Open Archives Metadata Set (OAMS), sendo alguns deles qualificados, para permitir a pesquisa de documentos nos arquivos;
- Uso de XML como sintaxe de transporte e representação;
- Utilização de um protocolo de transferência de dados comum.

Destes pressupostos nasce o OAI-PMH (Open Archives Initiative – Protocol for Metadata Harvesting), cujo o principal objectivo era ter a capacidade de transferir metadados dos arquivos fonte para os arquivos destino. Este protocolo nasce de uma prova de conceito feita com o protocolo Dienst[2], tendo evoluído até a versão usada actualmente, a versão 2.0 lançada em 2006. Nesta versão houve uma revisão do protocolo, sendo que a maior

[1] http://www.openarchives.org/OAI/OAI-organization.php

[2] http://www.cs.cornell.edu/cdlrg/dienst/protocols/DienstProtocol.htm

diferença passa pela forma como se aborda a informação, deixando de ser apenas documentos para passarem a ser recursos.

Como já foi mencionado anteriormente este protocolo tem como principal função a obtenção de metadados, sendo que não suporta qualquer tipo de pesquisa directamente sobre a informação obtida. A obtenção ou recolha de dados feita pelo OAI-PMH pode ser total ou selectiva, sendo que se pode obter informação como um todo ou apenas podem ser obtidos pequenas porções de informação, mediante as necessidades do service provider. Este processo é baseado em dois critérios, **sets** e **datestamps**, que podem ser aplicados em conjunto ou individualmente, sendo que no primeiro caso todos os dados pertencentes a um conjunto ou set serão obtidos, ao contrário do que acontece com a recolha por datestamp onde a é apenas recolhida informação que tenha sido alterada durante um período de tempo específico.

Os pedidos são tratados através dos métodos GET e POST do HTTP, sendo que as respostas a estes são sempre devolvidas em formato XML codificadas em UTF-8, tendo em atenção o esquema escolhido pela entidade para a codificação dos dados. Este protocolo tem a capacidade de tratar múltiplos tidos de metadados, mas o Dublin Core[16] passa por ser o imperativo para manter o propósito da interoperabilidade que este protocolo apresenta.
Para finalizar, em termos de estruturação de dados ao nível dos repositórios, o OAI-PMH lida com três tipos de estruturas de informação:

- Resources (recursos) – informação a partir da qual são gerados os metadados utilizados pelo OAI-PMH;
- Item – entidade mais abstracta do OAI-PMH, sendo o ponto de entrada para a disseminação da informação de um resource;
- Record (registo) – contém os metadados de um item, codificados num formato específico de XML (Dublin Core, MARCXML[17], METS[18]).

3.5 Framework de recolha de metadados

Com base no protocolo supracitado projectos como o REPOX foram desenvolvidos para lidar com a recolha automática de metadados. Esta tecnologia é amplamente utilizada e direccionada para lidar com os problemas que envolvem as bibliotecas e arquivos digitais.
Assim "o REPOX é uma implementação do Conceito de Repositório de Metadados"[11], que fornece uma plataforma de funcionamento aberta, sem o recurso a qualquer tipo de tecnologia proprietária no seu processo de preservação de dados.

Este sistema faz a gestão de várias colecções de metadados, provenientes de entidades diversas, sendo que cada uma delas está ligada a uma interface de fonte de dados, que serão responsáveis pela obtenção dos registos

[16] Dublin Core Metadata Initiative(DCMI), http://dublincore.org/

[17] MARC 21 XML Schema, http://www.loc.gov/standards/marcxml/

[18] Metadata Encoding and Transmission Standard (METS) Official Web Site, http://www.loc.gov/standards/mets/

gerados e verificação dos mesmos antes sua integração no sistema REPOX. Estas verificações passam essencialmente por codificar os registos obtidos no esquema XML que tenha sido adoptado pela entidade que o gera, sendo que a obtenção dos mesmos pode ou não ter uma periodicidade obrigatória definida.

Arquitecturalmente toda a estrutura do REPOX[12] pode ser observada na imagem em baixo, sendo que o principal destaque de interesse desta infra-estrutura é o gestor de repositório. Este é implementado em Java EE[19] e tem como principal funcionalidade a recolha e gestão de dados provenientes das diferentes interfaces.

O REPOX possui várias interfaces para a recolha de dados sendo elas o HTTP, FTP, o OAI-PMH e o Z39-50. De todas estas interfaces a mais utilizada é o OAI-PMH, visto este protocolo ser um requisito comum requerido para a partilha de dados por parte das organizações que assim o desejem.

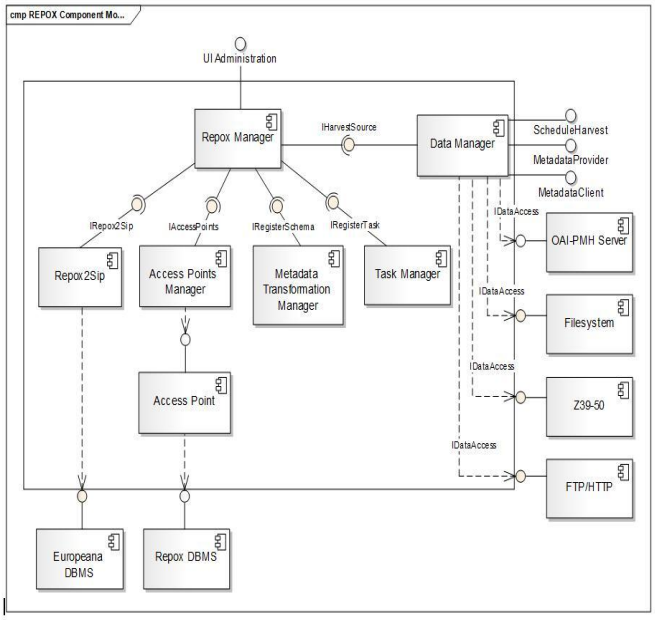

Figura 1- Arquitectura de componentes da framework

Tendo em conta os desafios e as tecnologias apresentados anteriormente foi possível definir um conjunto de objectivos e de requisitos para este projecto.

[19] http://www.oracle.com/technetwork/java/javaee/overview/index.html

3.6 Objectivos para o sistema de obtenção e agregação de dados

Como é possível observar, através da leitura deste documento, não existe uma tecnologia desenhada directamente para lidar com os problemas da recolha dos objectos referenciados pelos metadados, sendo que foi necessário o desenvolvimento desta, para que fosse possível definir todos os objectivos para o sistema de obtenção e agregação de dados. Assim os objectivos definidos foram:

- Gestão da informação relativa aos Data providers e aos seus servidores OAI-PMH, os seus data sets e os seus schemas descritivos de metadados.
- Recolha dos objectos referenciados pelos metadados provenientes dos servidores de OAI-PMH.
- Monitorização das recolhas efectuadas para verificação da qualidade das mesmas.
- Gestão do armazenamento dos objectos recolhidos mediante os seus data sets.

Estes objectivos permitem a definição de um serviço que efectua a recolha de objectos referenciados por metadados, desde que a sua origem esteja definida nos parâmetros acima mencionados.

3.7 Requisitos para o sistema de obtenção e agregação de dados

Os requisitos provêm dos objectivos e do âmbito em que estão inseridos. Tendo em conta os objectivos definidos na secção anterior e análise feita ao sistema que se deseja implementar os requisitos que foram obtidos são os seguintes:

- Recolha dos objectos referenciados: Tem de ser possível recolher os objectos referenciados em qualquer data set e em qualquer schema, desde que este esteja devidamente representado em XML;
- Gestão das recolhas: Capacidade de adicionar, alterar e remover as informações das recolhas de objectos referenciados;
- Monitorização das recolhas: Capacidade de monitorização em tempo real e após o fim da recolha de informação relevante e relativa à mesma;
- Gestão dos dados obtidos das recolhas: Capacidade de alocação e de remoção dos dados relativos às recolhas feitas.

4. Solução e a sua implementação

Neste capítulo irá ser demonstrada a solução implementada partindo dos objectivos e dos requisitos definidos no capítulo 3. Será apresentada uma visão global do sistema, a arquitectura pensada como solução para as questões levantadas durante a tese, terminando com a apresentação da implementação da solução desenhada.

4.1 Visão do sistema

FullText Harvester é uma aplicação Web, desenvolvida principalmente em Java de forma a ser transversal aos vários sistemas operativos, que tem como principal objectivo a recolha de conteúdos referenciados por metadados. Para se conseguir este objectivo esta tecnologia foi desenvolvida assente em dois pressupostos, referentes à origem dos metadados que contêm as referências para os conteúdos. Estes são que os metadados são recolhidos por OAI-PMH, protocolo que tem sido alvo de estudo ao longo da escrita deste documento, ou que estes são obtidos previamente de outra forma que não a utilização do protocolo, como por exemplo por FTP/HTTP, sendo guardados no sistema de ficheiros de uma máquina e que o FullText Harvester irá ler directamente destes para proceder à recolha dos conteúdos referenciados.

Com esta tecnologia é possível, não só efectuar a recolha total de todos os objectos presentes num conjunto de registos de metadados, como recolher apenas os registos mais recentes ou apenas os conteúdos alterados, de forma a reduzir assim o tempo despendido para recolher toda uma colecção, e que a quantidade de informação transferida seja reduzida ao mínimo. É também possível parar em qualquer altura parar a recolha de conteúdos, assim como alterar ou remover as definições referentes a cada conjunto de registos ou meramente destruir todos os conteúdos até agora recolhidos para cada conjunto de registos.

Esta aplicação possui ainda a flexibilidade de poder ser extensível a outras tecnologias, pois possui um conjunto de comandos REST, que permitem a invocação da mesma de modo a puder ser utilizada em múltiplos âmbitos que envolvam a recolha de objectos contidos em XML.

4.2 Arquitectura do sistema

A arquitectura do sistema, foi pensada com o objectivo de cumprir todos os objectivos e requisitos definidos nas secções 3.6 e 3.7. Esta é constituída por um conjunto de elementos que lhe permitem ter a flexibilidade e a escalabilidade necessária para garantir que, independente da forma como os metadados são apresentados e a localização dos objectos nos ficheiros recolhidos, é sempre possível efectuar recolha dos mesmos de forma eficiente, produzindo desta forma uma solução capaz de colmatar a necessidade tecnológica apresentada nesta tese. Assim o sistema é constituído por:

- Interface gráfica que facilite a utilização da tecnologia (GUI);

- Manager que gere todas os pedidos dos utilizadores referentes ao controlo das recolhas;
- Harvester que promove as recolhas e a sincronização de objectos digitais
  - OaiHarvester que trata dos pedidos que venham pelo protocolo OAI-PMH
  - FolderHarvester que trata de todos metadados que foram recolhidos de outras formas;
- DownloadFiles que efectua todo o tipo de recolha de objectos referenciados em metadados guadando essa informação no FullText Repository;
- HarvesteLog que gere toda a informação produzida ao longo das recolhas de forma a gerar relatórios que repliquem de forma precisa os acontecimentos inerentes as recolhas.

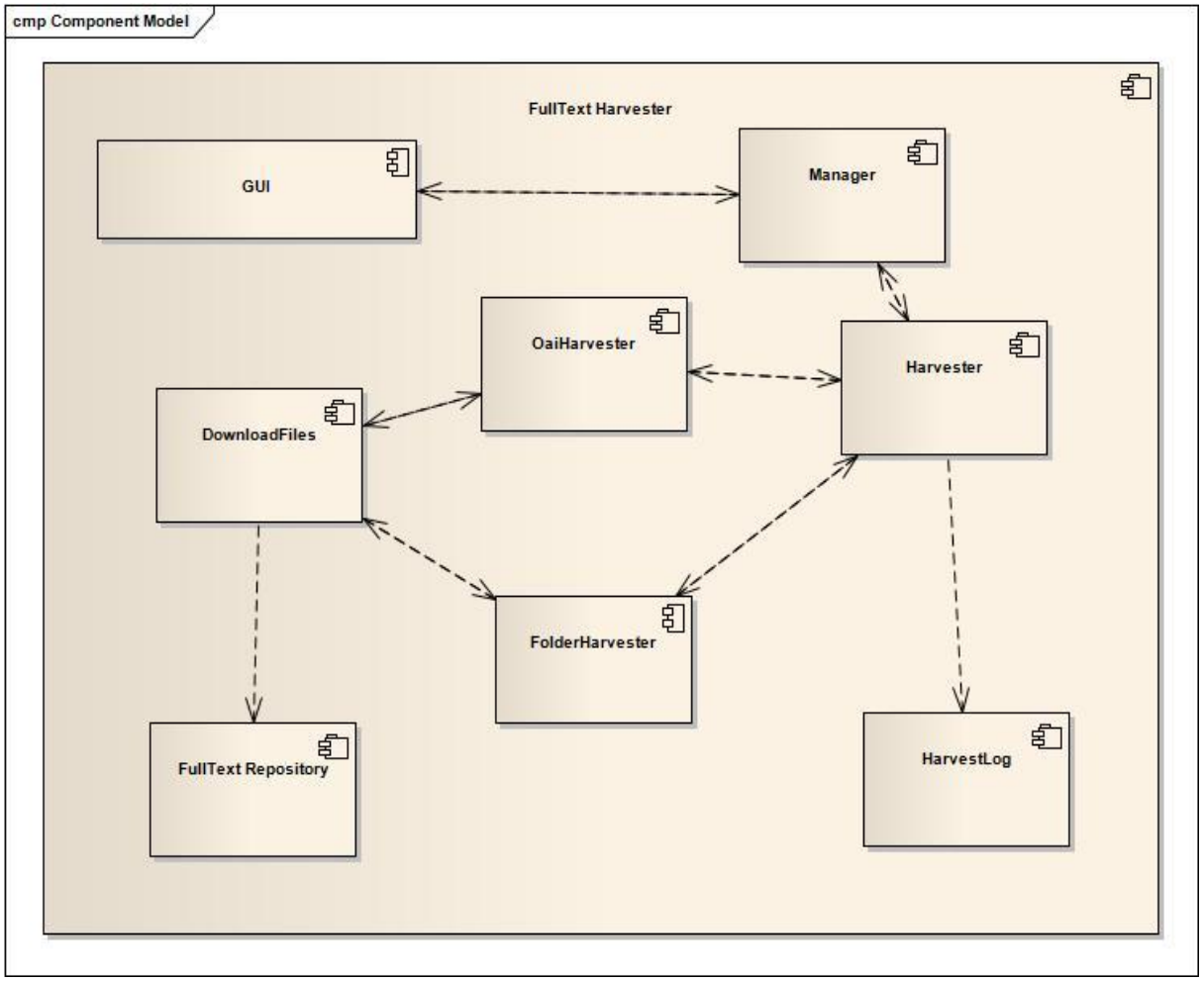

Figura 2- Arquitectura do FullText Harvester

4.3 Funcionamento da tecnologia Full-Text

Esta secção tem o objectivo de demonstrar o funcionamento da tecnologia Full-Text, focando-se no desenho desta e das soluções utilizadas para cada desafio.

4.3.1 Recolha de objectos referenciados

Para que seja efectuada uma recolha de objectos referenciados em metadados, a primeira acção que tem de ser feita pelo utilizador, é a definição das configurações necessárias para que o Full-Text Harvester saiba onde efectuar a recolha, qual o conjunto de registos que deve analisar e o formato em que vêm os metadados, a localização para onde irá ser feita a recolha dos objectos e o Xpath que irá ser utilizado para definir a localização dos objectos nos metadados.

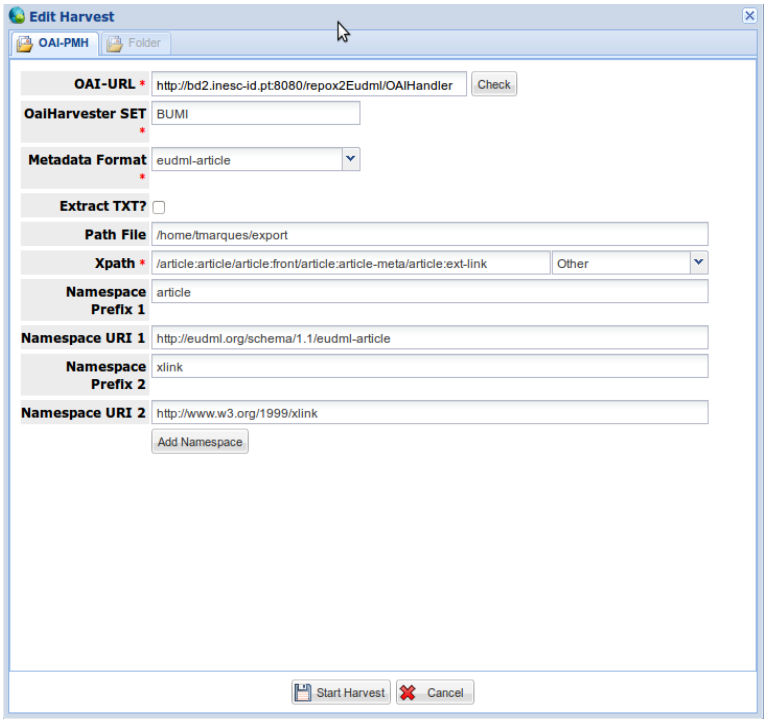

Figura 3- Imagem de configurador de recolhas de dados

Todos estes parâmetros definidos são utilizados pelo componente mais importante de toda a arquitectura do sistema, que é o Manager. O Manager é responsável pela gestão de todo o funcionamento associado às threads, que são executadas a cada recolha de objectos, sendo mantida uma lista de todas as threads e dos seus estados associados, consoante se estas foram concluídas com sucesso, com erros, ou se ainda se encontram em execução. Cada thread é representada por um objecto de nome threadInfo. Este objecto contém informação importante sobre cada thread como:

- O identificador único;
- O tipo de thread (ThreadOai ou ThreadFolder);
- A localização de onde são armazenados os objectos obtidos;
- O xpath utilizado para analisar o ficheiro de metadados;
- O número total de registos presentes nos metadados;
- O número de ficheiros obtidos.
- O tamanho total ocupado pelos objectos recolhidos.

Consoante o tipo de recolha que irá ser efectuada, seja esta a partir do protocolo OAI-PMH ou partindo de metadados recolhidos de outra forma, são criados tipos de thread em consonância com a fonte da informação recolhida, sendo criado uma ThreadOai ou uma ThreadFolder tendo em conta os casos analisados.

Estes objectos são semelhantes a ThreadInfo partilhando todos os campos do mesmo adicionando apenas campos específicos necessários para cada tipo de thread recolha de dados.

Concluída a criação das estruturas de informação base para o funcionamento da aplicação, estas são alocadas a threads para que possa haver recolha de informação de múltiplos conjuntos de registos simultaneamente. Esta recolha de registos é feita no componente chamada Harvester, mais especificamente no OaiHarvester, devido ao desenvolvimento deste projecto se ter focado principalmente no protocolo OAI-PMH.

Para cada recolha é guardada o máximo de informação que é possível sobre a mesma sendo esta a seguinte:

- Número de registos a percorrer;
- Número de registos percorrido;
- Número de registos apagados;
- Número de ficheiros obtidos;
- Número de ficheiros que não foi possível obter;
- A quantidade de espaço ocupada em disco pelos ficheiros obtidos;
- Uma listagem dos ficheiros falhados e das razões que levaram a falha dos mesmos;
- O último registo obtido;
- O tempo de início de execução da recolha;
- O tempo de fim da execução da recolha;
- Uma listagem da duração da recolha efectuada para cada registo.

Todas estas informações são vitais para o controlo do funcionamento da aplicação e análise do progresso das recolhas, como são também pontos-chave para verificação da qualidade e completude das recolhas efectuadas.
Para cada conjunto de registos, de onde se pretenda obter os objectos referenciados, é feita uma chama ao protocolo OAI-PMH com o nome do conjunto e o tipo de metadados que este contém. Este vai retornar uma lista de registos que irá ser percorrida, elemento a elemento, de forma a obter todos os conteúdos desejados. Para cada registo é obtido o identificador que o acompanha e consecutivamente, extraída uma lista de todos os elementos

que correspondam ao Xpath, fornecido pelo utilizador, para assim obter os conteúdos desejados.

Através do elementos que são recolhidos da compilação do Xpath sobre o xml obtido dos registos, é possível criar um perfil dos objectos que irão ser recolhidos, tendo para isso de se percorrer os atributos que cada elemento possua de forma a extrair o máximo de informação possível. Assim, para aglomerar esta informação, foi criada uma estrutura de dados de nome DataInfo que possui os seguintes conjuntos de informação:

- Formato dos objectos que irão ser recolhidos;
- O endereço do objecto a ser recolhidos;
- O conjunto de registos a que pertence;
- O mime type;
- A extensão do objecto;
- A localização do ficheiro obtido;
- As permissões associadas aos ficheiros a ser recolhidos (recolha, correcção e partilha).

Para cada objecto existente no registo é criado um objecto do tipo supracitado e adicionado a uma lista, que irá ser posteriormente percorrida de forma a proceder à recolha dos objectos que lhe estejam implícitos.

Terminada a criação das estruturas de dados referentes à informação dos objectos é chegado o momento de fazer a recolha dos mesmos.

Esta recolha é efectuada por um componente de nome DowloadFiles, que irá retornar ao OaiHarvester, em caso de sucesso a localização do objecto obtido no sistema de ficheiros, ou caso de falha uma, excepção com a respectiva falha para que esta informação seja posteriormente escrita no registo de execução da aplicação.

O componente DownloadFiles subdivide-se em dois componentes. O HarvestSimpleFile ou o HarvestHtml. Para o HarvestSimpleFile são enviados os objectos que não requerem qualquer tipo de análise da sua localização para que possam ser efectuadas as recolhas dos mesmos. É efectuada uma ligação ao url extraído dos metadados, tendo a preocupação as redirecções que este possa impor. É também definido um timeout para a obtenção de resposta por parte do servidor, para que a continuidade e fluidez do processo de recolha não seja afectada. Em caso da excepção por timeout ser alcançada, este erro é reportado no relatório produzido na recolha dos objectos. No caso do HarvestHtml, como o próprio nome indica, são enviados os objectos que é necessário ser feita uma análise a páginas HTML, de forma a conseguir recolher os objectos que estão referidos nos metadados. Esta análise é efectuada verificando todos os elementos da página que contenham alguma informação, relativa ao objecto que se pretende recolher. Tendo em consideração o âmbito principal desta tese, que trata da obtenção e sincronização de objectos referenciados em metadados partilhados por bibliotecas e arquivos digitais, especificamente fulltext, o processo de recolha dos ficheiros foi de alguma forma orientado para o tratamento de dados dos tipos PDF, TXT, DOC, XML e HTML.
Após a recolha dos objectos presentes em cada registo, tenha sido esta com ou sem sucesso é gerado um relatório com os resultados produzidos pela aplicação. Este é invocado pelo OaiHarvester e tem o nome de HarvestLog.

Este componente tem a função de criar o registo que assinala as ocorrências em cada execução da thread relativa a um determinado conjunto de registos, sendo por isso tantos registos quantas novas execuções sobre uma determinada thread sejam pedidas. Existe ainda um outro relatório que mantém o estado global de uma thread, mantendo os aspectos mais relevantes relativos ao estado da mesma.

Um exemplo de um relatório ou log de cada execução pode ser visto na imagem seguinte.

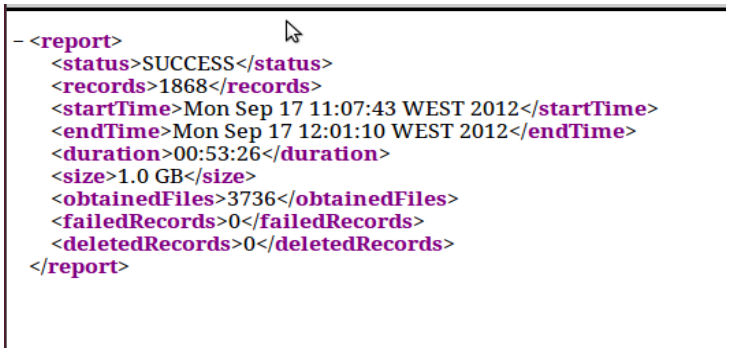

Figura 4- Relatório recolha com sucesso

Aqui é possível ver um exemplo de relatório de execução relativo a um conjunto de registos específico. É facilmente observável que esta execução correu sem erros, onde foram obtidos 3736 ficheiros a partir de 1868 registos num período de cerca de 53 minutos.

Na imagem seguinte é possível observar um exemplo onde ocorreram erros.

Figura 5- Relatório recolha com erros

Neste caso é contabilizado o número de erros ocorridos e é indicada a razão o identificador do registo que contem o erro e ainda o url utilizado para se tentar obter o objecto. Esta informação, como foi o caso, é de

extrema importância para quem publica os objectos pois como pode ser observado o url utilizado é inválido e permite assim que sejam reportados erros para uma maior eficácia em termos de recolha de informação.

Um exemplo do log total relativo a um conjunto registos sobre o qual tenha sido feito uma recolha de objectos encontra-se abaixo.

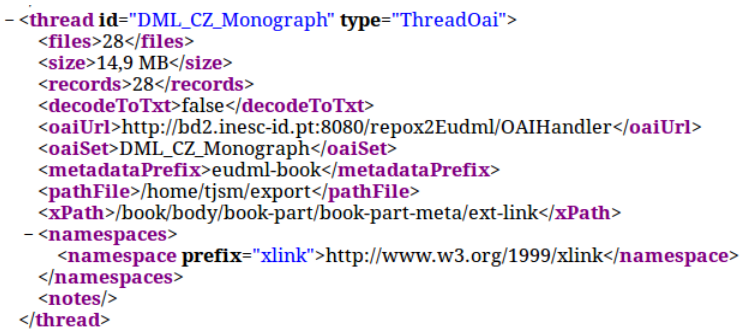

Figura 6- Relatório geral de um set

Esta informação que pode ser observada, pretende ser mais focalizada na evolução global do conjunto de registos analisados, sendo que contem um estado minimalista da evolução da thread, com a contagem dos ficheiros e dos registos que já foram analisados e a quantidade de informação já recolhida e o resto da informação presente é relativa às configurações específicas para este conjunto, como o nome do grupo de registos analisado, o tipo de metadados, a localização de para onde os objectos recolhidos estão a ser guardado e ainda o Xpath utilizado para obter a informação sobre os conteúdos a recolher.

4.3.2 Sincronização de ficheiros

Sendo este um sistema que recolhe grandes quantidades de informação é importante que estas se mantenham actualizadas com as suas fontes, pois uma obtenção de novos objectos é uma tarefa morosa que não compensa em casos de pequenas alterações efectuadas em poucos ficheiros. Para garantir esta premissa, foram utilizadas na tecnologia desenvolvida três técnicas complementares, que permitam garantir a sincronização dos conteúdos.

1. Utilizar o OAI-PMH para obter os registos mais recentes a partir de uma determinada data.
2. Verificar para todos os registos a última data de modificação de cada objecto.
3. Verificar o checksum de todos os objectos contidos nos registos.

O primeiro processo aproveita uma capacidade inerente ao protocolo OAI-PMH de permitir obter registos de um determinado conjunto de registos, que tenham sido alterados ou actualizados a partir de uma determinada data. Isto permite reduzir de forma significativa o número de registos analisados e ter assim um ponto de partida para apenas fazer uma nova recolha dos objectos referenciados por estes registos alterados.

Infelizmente, a actualização dos registos nem sempre reflecte o estado actual dos objectos que estes referenciam, sendo estes últimos actualizados sem que haja uma actualização da informação relativa ao registo. Por esta razão

é necessário verificar o estado dos mesmos e caso estes tenham sido actualizados, para efectuar a respectiva actualização dos dados recolhidos. Para isso são utilizadas outras duas técnicas mencionadas no início da secção. Caso haja, no momento da actualização, informação sobre a data de modificação do ficheiro este é recolhido, desde que a data de modificação seja posterior à da última recolha. Em caso de inexistência desta informação é feita uma recolha do objecto de forma a gerar o checksum do mesmo sendo comparado com o checksum do ficheiro já existente em sistema. Caso sejam diferentes é guardada a versão anterior e a nova versão toma o lugar da anterior, garantindo assim a actualização dos ficheiros.

Todas estas informações são validadas com base num relatório criado aquando da obtenção dos objectos na primeira recolha, e actualizado a cada nova obtenção de dados efectuada. Este pode ser considerado uma espécie de radiografia dos registos onde é contida a informação sobre cada registo e de todos os objectos por estes referenciados. Este relatório está detalhado da seguinte forma:

- Identificador do registo;
- Objectos
  - Data de modificação
  - Checksum
  - Link utilizado para o download
  - Espaço ocupado por cada objecto recolhido.

A existência desta estrutura de informação permite acelerar o processo de actualização, pois reduz o tempo de procura e obtenção da informação necessárias para efectuar as validações que são mencionadas ao longo desta secção.

4.3.3 Recuperação de falhas na obtenção dos objectos digitais

À semelhança do que acontece na actualização dos objectos recolhidos é importante verificar se registos em que tenham sido encontrados erros aquando da sua recolha, se estes já foram corrigidos e se já é possível efectuar a recolha dos mesmos sem ter de voltar a recolher todo o conjunto de registos. Com esse objectivo em mente utilizando os registos criados nas recolhas de objectos, os erros são colocados numa lista que irá invocar, para cada registo, uma chamada ao protocolo OAI-PMH de forma a criar uma nova lista, contendo apenas os registos que falharam e com todas as suas informações. Esta lista é analisada pelo Harvester e é efectuada a recolha dos objectos que falharam, caso seja possível, pois por cada registo com objectos que falharam é adicionada informação ao relatório, mantendo assim a consistência dos dados sobre o conjunto de registos.

4.3.4 Gestão de processos pela interface gráfica

Para ajudar tanto no desenvolvimento, como para facilitar a utilização da solução desenvolvida, foi criada uma pequena interface gráfica que é possível ver e comandar todas as operações que possam ser executadas sobre os conjuntos de registos que contêm metadados. Nesta interface é possível identificar duas secções muito importantes. A primeira é relativa às recolhas, onde é possível verificar que contem a informação mais básica sobre cada recolha, sendo esta os tempos de início e de fim da recolha e o estado da mesma, sendo estes de sucesso, erro, cancelado ou em recolha. Estes estados são visíveis na imagem abaixo, no campo Harvest Status, sendo claro que o sucesso é representado pelo símbolo verde, o erro com o vermelho, o cancelado pelo símbolo azul, que representa um utilizador que efectuou o cancelamento da recolha e finalmente o estado a recolher que não está presente na imagem mas é especificado por uma barra de progresso com a percentagem do mesmo.

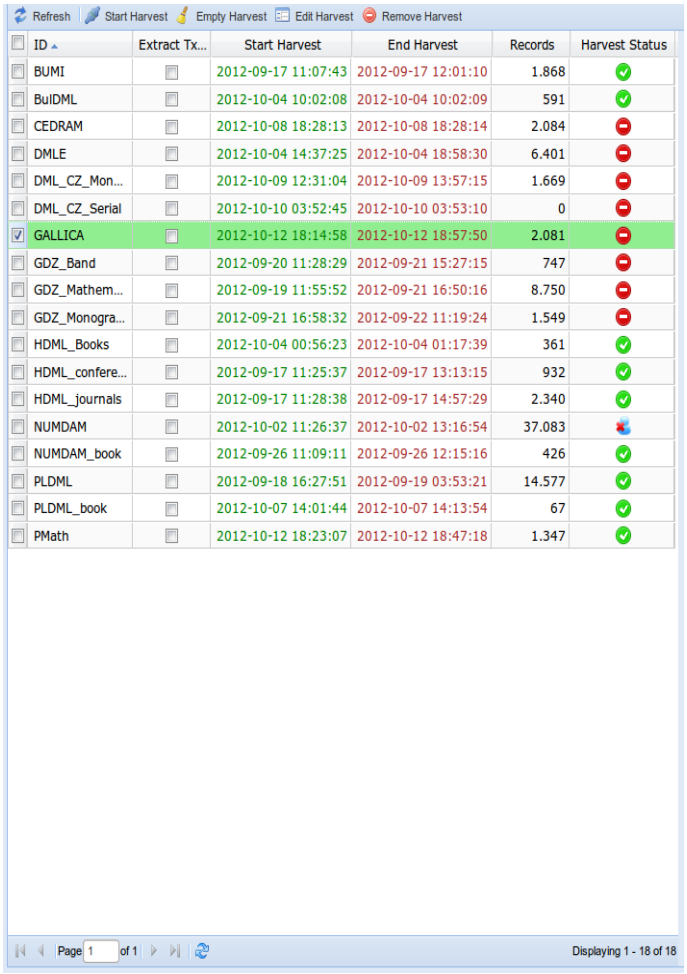

Figura 7- Imagem da secção de recolhas

O controlo das acções é permitido pelos botões se encontram na parte superior relativa às recolhas. Estes correspondem ao pedido para início de uma recolha (Start Harvest), onde tem várias acções associadas. Se a recolha estiver vazia efectua uma nova recolha. Caso esta tenha sido concluída com sucesso tenta fazer uma actualização da recolha. Caso esta tenha terminado com erros tenta efectuar a recuperação destes mesmos erros

para tentar que a recolha atinja o estado de sucesso. O botão de Empty Harvest efectua a remoção de todos conteúdos já recolhidos até ao momento mas mantendo todas as configurações necessárias para efectuar uma nova recolha em qualquer altura. O botão de Edit Harvest permite a reconfiguração da recolha permitindo alterar qualquer campo dos que foi mostrado na figura 3. Finalmente o botão Remove Harvest remove todos conteúdos, relatórios e configurações de uma recolha, sendo esta apagada da interface.

A segunda secção e de relevante importância está representada na figura seguinte. Esta permite essencialmente mapear o máximo de informação que possa ser importante para o utilizador a partir dos relatórios gerados ao longo da execução da recolhas, sendo possível observar o estado actual da recolha, o que foi feito na última recolha e ainda é dada a possibilidade de ver todos os relatórios sobre as recolhas feitas até ao momento.

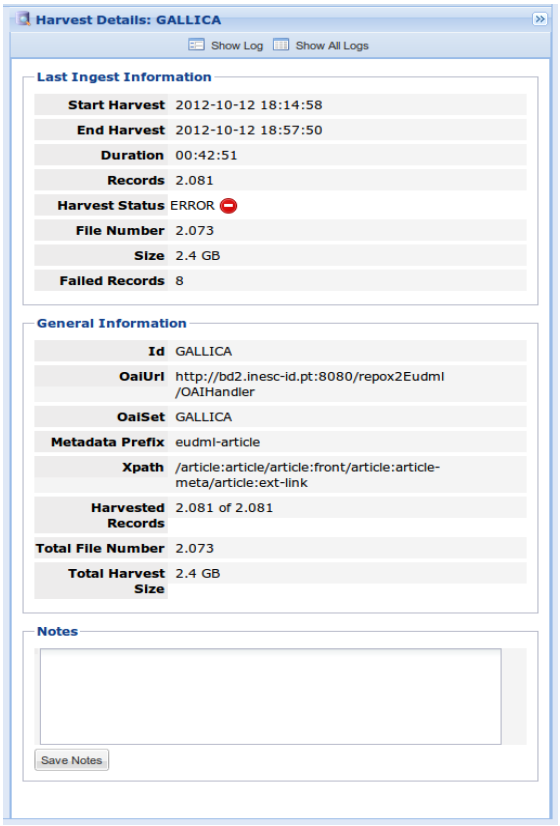

Figura 8- Imagem dos relatórios relativos às recolhas

É também possível para o utilizador utilizar o campo Notes, onde é possível anotar qualquer informação que possa ser relevante para a recolha. Finalmente caso se deseje ver o xml relativo ao estado da recolha é possível clicando símbolo referente à recolha no campo Harvest Status, ou clicando no botão no topo que diz Show Log, sendo redireccionado para uma página onde irá ser apresentado o relatório referente à recolha seleccionada.

Esta imagem é referente apenas ao caso de uma recolha concluída, pois caso esta esteja em execução as informações são actualizadas em tempo real, sendo os botões presentes no topo da interface substituídos por um botão que permite o cancelamento da execução da recolha.

5. Avaliação dos resultados

Neste capítulo irá ser feita uma avaliação da solução implementada, utilizando para isso cenários reais de utilização de interoperabilidade de bibliotecas digitais. Esta avaliação foi conseguida por execução de processos que testam toda a capacidade e funcionalidade da solução desenvolvida, desde a recolha de objectos referenciados passando pela sincronização de dados e terminando tentativa de obtenção de objectos cujos os registos contêm erros.

5.1 Metodologia utilizada na avaliação

Para efectuar a validação da tecnologia desenvolvida, foi utilizada a framework REPOX, que é utilizada em vários cenários como a Europeana, TEL, EuDML ou o SHAMAN. O principal cenário utilizado para o estudo foi o do projecto EuDML, pois é actualmente o que apresenta a mais elevada taxa de disponibilidade em termos de conteúdos prontos para serem recolhidos. Este projecto apresenta outros desafios em simultâneo, como a variedade de tipos de dados disponíveis para serem recolhidos e ainda a diversidade de formatações para cada conjunto de registos disponibilizados. Com este cenário de estudo é pretendido demonstrar toda a capacidade da solução desenvolvida apesar da heterogeneidade apresentada pelo mesmo.

5.2 Avaliação da recolha de objectos

Para o primeiro cenário de avaliação foram efectuadas várias recolhas de forma a possibilitar o máximo de informação possível para se poder trabalhar. Como é possível ver na imagem seguinte, não foi possível obter 100% de sucesso nas recolhas de objectos referenciados, pois existem casos de erros, alheios à solução desenvolvida que são apresentados nos relatórios gerados ao longo da execução dos mesmos.

Toda a informação recolhida perfaz um total que ronda os 200 GB. Este valor permite validar de alguma forma a capacidade de execução da solução, mas a maior validação é feita na base do número de registos que são armazenados pelo REPOX, que tem de corresponder ao mesmo analisado na recolha dos objectos referenciados. Esta assunção pode ser confirmada pela observação das figuras 9 e 10, analisando por exemplo o campo assinalado com o nome de GALLICA.

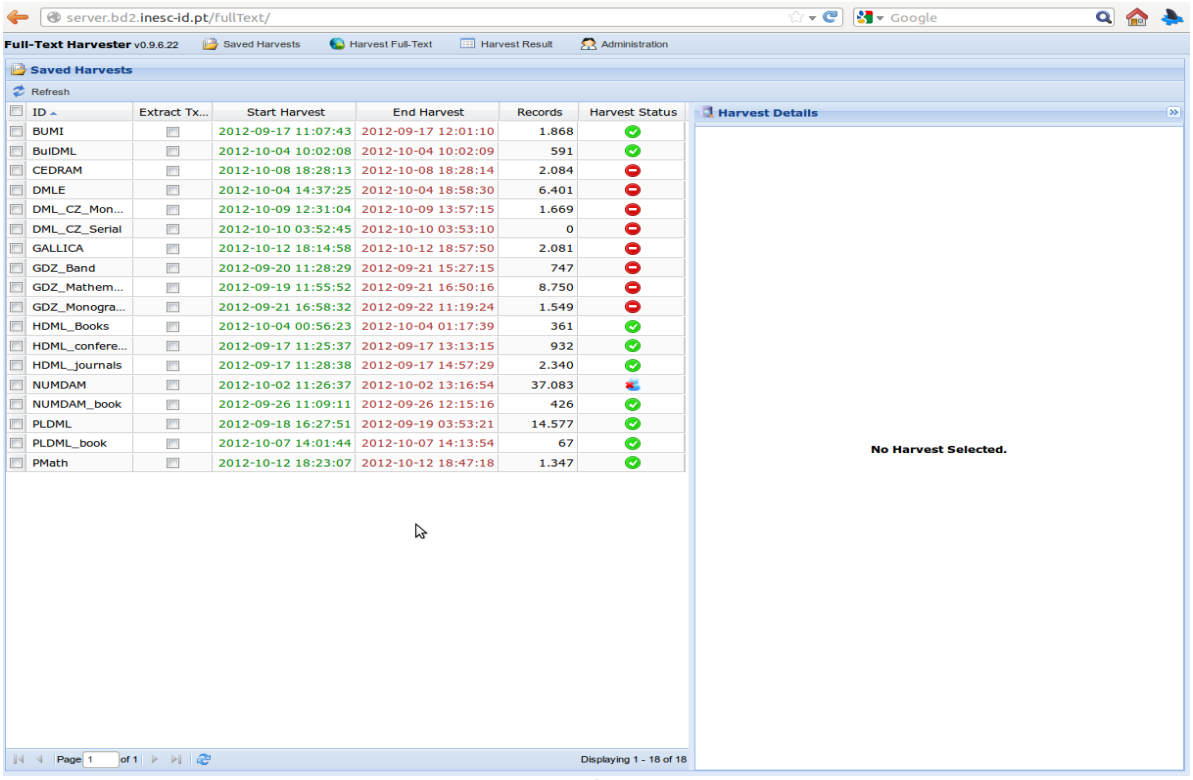

Figura 9- Harvester com um conjunto de recolhas efecutadas

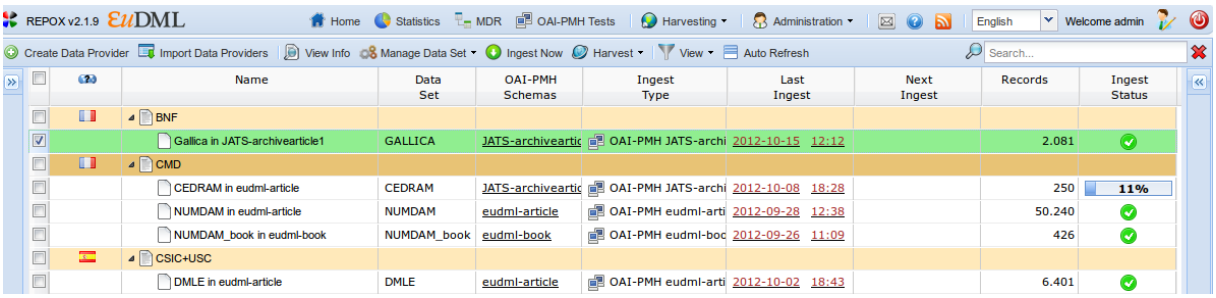

Figura 10- Repox EuDML (lista de registos)

5.3 Avaliação da actualização

Para efectuar a avaliação de um processo de actualização de dados foram utilizadas as fontes de informação do primeiro teste. Esta foi dividida em duas fases distintas. Na primeira foi feita uma avaliação da eficácia e eficiência de cada uma das técnicas apresentas na secção 4.3.2, individualmente. Na segunda fase foi avaliada a capacidade global da solução utilizando as três técnicas para efectuar a actualização de um conjunto de registos obtendo assim um comparativo entre uma recolha totalmente nova dos conteúdos e uma actualização dos registos alterados.

Avaliando as três soluções individualmente foi possível concluir que:

- A verificação dos registos alterados utilizando o protocolo OAI-PMH é uma solução muito rápida, que permite facilmente obter todos registos alterados, ficando o processo de sincronização limitado ao tempo de download dos objectos;
- A verificação da data de modificação dos objectos é processo rápido e eficaz, cobrindo todos os casos que a solução anterior apresenta e ainda descobre casos de registos que não foram actualizados, tendo sido apenas os objectos;
- A verificação dos checksum dos ficheiros é de todas a solução mais completa, mas também a mais dispendiosa, pois exige que seja feita a recolha dos objectos, sendo que o tempo de processamento deste é equivalente a qualquer recolha nova que seja feita.

Avaliando a solução implementada na tecnologia onde é feito um encadeamento das três técnicas foi possível verificar que:

- Em cerca de 90% dos casos o processo de actualização foi mais rápido que o tempo total de uma recolha de um conjunto de registos, devido à existência da data de modificação do objecto ou a existência de registos alterados, o que permite reduzir drasticamente o número de registos que têm de ser analisados pelo processo de actualização;
- Caso o valor da última data de modificação do objecto seja omissa ou inexistente ou não haja alterações nos registos que contêm os objectos o processo de actualização é tão ou mais lento que um processo de recolha total de todos os objectos, devido à quantidade extra de informação que é analisada e à simples obtenção de objectos para verificação de checksum dos mesmos, para em muitos casos este ser descartado.

5.4 Avaliação da recolha dos registos com erros

Para efectuar a avaliação do processo de recolha dos registos que tenham apresentado erros, foi novamente utilizada a informação gerada no primeiro teste. Assim tomando o exemplo seguinte onde o conjunto de registos de nome GALLICA, possui alguns erros para efectuar a verificação da correcção dos mesmos.

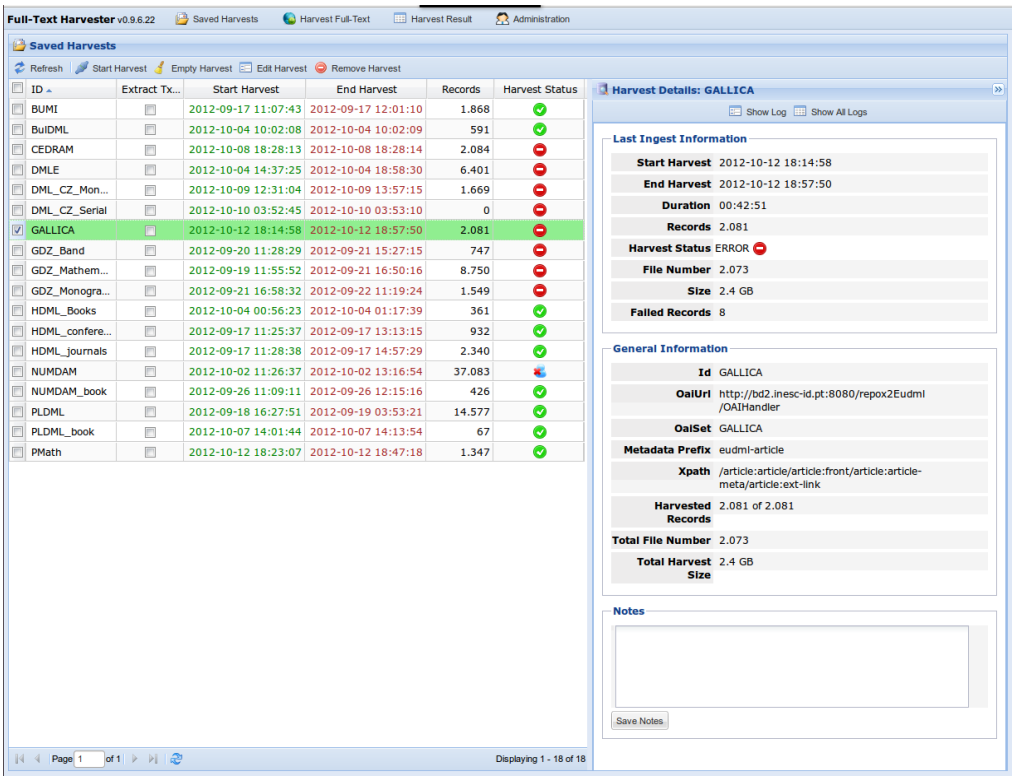

Figura 11 - Recolha com erros e relatório

Como é possível observar na figura 10 existem 8 erros pendentes em registos, cuja informação sobre os mesmos pode ser vista mais em detalhe, através do relatório seguinte.


```
- <report>
    <status>ERROR</status>
    <records>2081</records>
    <startTime>Fri Oct 12 18:14:58 WEST 2012</startTime>
    <endTime>Fri Oct 12 18:57:50 WEST 2012</endTime>
    <duration>00:42:51</duration>
    <size>2.4 GB</size>
    <obtainedFiles>2073</obtainedFiles>
    <failedRecords>8</failedRecords>
    <deletedRecords>0</deletedRecords>
  - <errors>
    - <error time="Fri Oct 12 18:29:07 WEST 2012">
      - <recordId>
          urn:eudml.eu:GALLICA:oai:eudml.mathdoc.fr:Gallica:JMPA_1853_1_18_A15_0
        </recordId>
      - <url>
          http://portail.mathdoc.fr/JMPA/PDF/JMPA_1853_1_18_A15_0.pdf
        </url>
        <cause>404 Not Found</cause>
      </error>
    - <error time="Fri Oct 12 18:34:03 WEST 2012">
      - <recordId>
          urn:eudml.eu:GALLICA:oai:eudml.mathdoc.fr:Gallica:JMPA_1859_2_4_A13_0
        </recordId>
      - <url>
          http://portail.mathdoc.fr/JMPA/PDF/JMPA_1859_2_4_A13_0.pdf
        </url>
        <cause>404 Not Found</cause>
      </error>
    - <error time="Fri Oct 12 18:21:16 WEST 2012">
      - <recordId>
          urn:eudml.eu:GALLICA:oai:eudml.mathdoc.fr:Gallica:JMPA_1846_1_11_A15_0
        </recordId>
      - <url>
          http://portail.mathdoc.fr/JMPA/PDF/JMPA_1846_1_11_A15_0.pdf
        </url>
        <cause>404 Not Found</cause>
      </error>
    - <error time="Fri Oct 12 18:33:19 WEST 2012">
      - <recordId>
          urn:eudml.eu:GALLICA:oai:eudml.mathdoc.fr:Gallica:JMPA_1856_2_1_A32_0
        </recordId>
      - <url>
          http://portail.mathdoc.fr/JMPA/PDF/JMPA_1856_2_1_A32_0.pdf
        </url>
        <cause>404 Not Found</cause>
      </error>
    - <error time="Fri Oct 12 18:34:30 WEST 2012">
```


Figura 12- Relatório com erros de um set recolhidos

Neste caso específico foi possível verificar que após se ter tentado recolher os registos com erro, os mesmo se mantiveram gerando um relatório semelhante a este em que apenas mudou a informação do número de registos analisado passando a ser oito, tantos quanto o número de erros.

Em outros testes semelhantes foi possível analisar que ou todos os registos eram recolhidos com sucesso ou que apenas parte deste era recolhido com sucesso, o que leva a concluir que este é um processo importante para a recolha de objectos referenciados, pois evita que seja feita um novo download de todos os objectos existente no conjunto de registos, melhorando a eficiência das recolhas e a eficácia com que estas são feitas.

5.5 Sumário

O resultado deste estudo sugere que a solução desenvolvida tem a capacidade e o potencial para efectuar, com sucesso, a recolha de objectos referenciados em metadados, como foi provado com o cenário real do EuDML, o que leva a crer que outros cenários suportados pela framework REPOX serão igualmente recolhidos com sucesso.

É também possível observar que a dimensão de muitos dos conjuntos de registos analisados, requer que, em caso de actualização de alguma da informação presente nos objectos recolhidos, estes também sejam actualizados, sendo de grande importância que a existência de um procedimento que cumpra este desígnio, algo que foi provado como estando funcional e sendo o mais eficiente possível mediante a forma de disponibilização dos objectos.

Finalmente é importante ressalvar a relevância do processo de tentativa de recolha de objectos, sobre os quais existe alguma impossibilidade de recolha dos mesmos, pois torna a recolha como um todo mais eficiente, evitando a necessidade de recolher novamente todos os objectos pertencentes a um conjunto de registos, de forma a validar a correcção ou a recorrência dos erros.

No próximo capítulo será feito uma análise e um resumo de tudo o que tem sido dito ao longo da dissertação, de forma a clarificar as contribuições e as limitações deste trabalho, mostrando ainda o que pode ser feito no futuro tendo em consideração este tema.

6. Conclusão e trabalho futuro

O aparecimento das bibliotecas e arquivos digitais criou a possibilidade de se partilhar com a comunidade a informação que cada uma destas possui, para que esta possa ser pesquisada e analisada para os mais variados fins e desejos. Com este objectivo em vista foi criado o REPOX, uma framework que funciona como agregador de metadados descritivos das informações que as bibliotecas e os arquivos desejem disponibilizar. Estas são publicadas de forma a facilitar a sua pesquisa por parte dos utilizadores, permitindo a estes encontrar as informações que desejem para depois se dirigirem as entidades responsáveis pela partilha dos dados para obtenção de informações mais detalhadas do que se deseje.

Com o objectivo de melhorar as pesquisas por parte dos utilizadores das bibliotecas e arquivos digitais foi proposto um novo desafio. Este passa pela obtenção e partilha de conteúdos, principalmente fulltext, ou seja, para além dos metadados descritivos obtidos na primeira fase deste projecto, nesta fase serão obtidos conjuntamente todos os conteúdos produzidos para que possam ser pesquisáveis e assim permitir uma maior e melhor qualidade de pesquisa.

Com este novo paradigma em vista deixa de ser possível manter apenas os protocolos de obtenção de dados que se encontram instituídos nas partilhas de dados entre bibliotecas digitais, sendo caso disso o OAI-PMH, que se revelou um sucesso em termos da obtenção dos metadados em diferentes localizações, mas que devido às suas características acaba por não ser suficiente para cumprir os novos objectivos apresentados.

Assim o objectivo desta dissertação passou pela procura da melhor solução que permita resolver o problema introduzido neste documento. Esta tinha o objectivo de lidar com vários tipos de dados e promover a recolha dos mesmos, sempre com o objectivo de reduzir custos de banda larga, processamento e tempo despendido aquando da actualização dos mesmos e ainda que seja escalável.

Como foi descrito no capítulo 2, existem várias soluções testadas que permitem efectuar a sincronização de ficheiros, mas não foi possível encontrar uma solução que lidasse com todos os desafios apresentados, nomeadamente com a sincronização de ficheiros cujos seus originais estejam alocados em servidores remotos, cujo o único acesso que existe é através de urls para os mesmos. Assim foi necessário efectuar uma análise mais detalhada do problema de forma a definir claramente quais os objectivos e os requisitos para resolver a questão proposta, como foi efectuado no capítulo 3.

Assim no capítulo 4 é possível analisar a arquitectura e funcionamento do sistema de recolha desenvolvido. Este dá a possibilidade aos utilizadores de efectuar recolhas de objectos digitais, referenciados em metadados,

mediante algumas simples configurações, que indiquem à aplicação a localização dos metadados a ser analisados, o formato dos mesmos e o Xpath que indicará a localização dos conteúdos nos metadados. Esta tecnologia permite que seja feita um acompanhamento e gestão em tempo real da evolução das recolhas, através da simples interface que foi criada para a utilização do mesmo. Tendo sido terminada a fase das recolhas de dados foi possível partir para a criação de uma solução que permitisse a sincronização dos objectos recolhidos, acção que revelou ser o maior desafio, pois a impossibilidade de acesso directo aos elementos recolhidos levou a uma abordagem para a garantia de sincronização que pode não ser a mais eficiente academicamente, mas que acabou por ser a mais eficaz em termos aplicabilidade em casos reais, devido as particularidades das recolhas. A solução introduzida é de extrema importância, pois a quantidade de dados recolhidos e o tempo despendido para algumas das recolhas exige que não seja feita uma nova recolha total por parte da informação, o que foi confirmado com sucesso na maioria dos casos estudados. Em complemento a este processo de actualização, foi introduzido um sistema que permite apenas tentar recolher os registos que tenham tido algum tipo de erro aquando da recolha dos seus objectos referenciados, o que mais uma vez revelou ser uma solução importante para a diminuição da quantidade de informação necessária para manter os dados o mais actualizado possível.

No capítulo 5 foi feita a avaliação da tecnologia desenvolvida, utilizando casos de utilização real, garantindo desta forma que os resultados são os mais precisos e demonstram a real utilização da mesma. Durante a realização dos testes foi possível observar que a solução desenvolvida efectua com sucesso a recolha de objectos referencia em metadados, especialmente fulltext que foi o principal tipo de objecto recolhidos. Em termos de sincronização esta solução apresentou uma eficácia que depende do número de objectos que foi actualizado, visto que a recolha dos objectos actualizados consiste na obtenção total do objecto. De qualquer das formas é possível concluir que a solução implementada é adequada ao cenário em que se integra, pois o número de objectos que é normalmente actualizado é bastante reduzido, sendo assim possível obter elevadas taxas de eficácia e eficiência na actualização dos objectos.

Em conclusão, foi possível verificar que todos os objectivos propostos no capítulo 3 foram alcançados o que permite validar que foi desenhada uma solução para suportar a recolha e sincronização de objectos digitais referenciados em metadados, nomeadamente em cenários reais suportados pelo REPOX como é o caso do EuDML.

6.1 Trabalho Futuro

Uma das limitações do trabalho desenvolvido passou por não ter sido desenvolvida uma solução que promova a partilha dos metadados e dos objectos referenciados pelos mesmos entre Service Providers que utilizem a tecnologia REPOX. Isto irá permitir que a partilha de informação seja efectua mais rapidamente e que as actualizações possam ser efectuadas utilizando soluções mais direccionadas para a sincronização de ficheiros.

A limitação supracitada pode ainda tentar aproveitar as estruturas de dados desenvolvidas ao longo da execução desta tese, tentando de alguma forma utilizar o relatório gerado aquando da recolha dos dados, que contem a

informação descritiva do registo e dos objectos por ele referenciados, o que em conjugação com alguma das tecnologias estudadas no capitulo 2 ou eventualmente sobre o próprio OAI-PMH, poderão apresentar uma solução de máxima eficácia para as recolhas feitas em cenários de bibliotecas e arquivos digitais.

Outra funcionalidade que pode ser implementada e que poderá trazer valor é a criação da funcionalidade para efectuar agendamentos de recolhas ou de actualizações de objectos referenciados. Este caso específico será útil em caso indisponibilidade imediata de algum conjunto de registos e esta funcionalidade permitirá agendar recolhas diminuindo assim a necessidade de interacção com o utilizador.

Finalmente poderá ser introduzida uma funcionalidade que permita a recolha de objectos, tendo em conta o conteúdo dos mesmos, pois neste momento a recolha dos dados é feita tentando procurar todos os objectos presentes a partir de um Xpath, e esta opção iria permitir a especialização de agregadores de dados em tipos de dados específicos.

7. Referências bibliográficas

[1]Carl Lagoze and Herbert Van de Sompel. The open archives initiative: building a low-barrier interoperability framework. In JCDL '01: Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries, pages 54–62, New York, NY, USA, 2001. ACM.

[2]Van de Sompel, Lagoze. D-Lib Magazine February. 2000, Vol. 6 Number 2.Consultado em: http://www.dlib.org/dlib/february00/vandesompel-oai/02vandesompel-oai.html

[3]SyncML White Paper:Building an Industry-Wide Mobile Data Synchronization Protocol

[4]Andrew Tridgell and Paul Mackerras. The rsync algorithm. The Australian National University

[5]David Rasch and Randal Burns. In-Place Rsync. File Synchronization for Mobile and Wireless Devices. Department of Computer Science Johns Hopkins University

[6]John Langford. Multiround Rsync. 2001

[7]Benjamin C. Pierce and Jerome Vouillon. What's in Unison? A Formal Specification and ReferenceImplementation of a File Synchronizer. Department of Computer & Information Science. 2004

[8]A.-M. Kermarrec, A. Rowstron, M. Shapiro, and P. Druschel. The icecube approach to the recon-ciliation of divergent replicas.

[9]L. Veiga and P. Ferreira. Semantic-Chunks a middleware for ubiquitous cooperative work. In Proceedings of the 4th workshop on Reflective and adaptive middleware systems, page 6. ACM, 2005.

[10]S. Zachariadis, L. Capra, C. Mascolo, and W. Emmerich. XMIDDLE: A Data-Sharing Middleware for Mobile Computing. 2002

[11]Freire, Manguinhas, Borbinha, REPOX: Uma infra-estrutura XML para a PORBASE, Lisboa

[12]Freire, Manguinhas, Borbinha. Metadata Spaces: the concept and a case with REPOX, International Conference on Asian Digital Libraries. 2006

[13]Ian H. Witten, David Bainbridge, David M. Nichols How to Build a Digital Library, pages 7-8, Second Edition.